НЕ ПРАВЕТЕ КАТО ТЯХ, ЗА ДА НЕ СТАВАТЕ ЗА СМЯХ

Уважаеми читатели,

На ваше разположение са шедьоврите на вашите колеги, получени на изпитите по дисциплината „Компютърни архитектури“. Тази рубрика е създадена с мисълта, че разумните хора могат да се учат и от грешки (по-добре от чуждите, вместо от своите). Ако някой отговор не ви се стори достатъчно смешен за да попадне в тази рубрика, прочетете отново въпроса, на който той е даден, и не забравяйте, че всички термини се обсъждат от гледна точка на жаргона на науката Информатика. Всички отговори се публикуват без каквато и да е редакторска намеса от моя страна, както и без връзка с подреждането им във въпросника за изпит. В редки случаи за по-голяма яснота след отговора **в зелено** е даден и моят **пояснителен коментар**.

Във втория том на рубриката се публикуват само шедьоврите, получени след 100 издание. Статистиката в края на изданието се води за двата тома едновременно.

Включването в рубриката „Не правете като тях, за да не ставате за смях“ (и резил) **е безплатно и анонимно**. **Рецидивите се наказват с обявяване на авторските права** на студента.

За да се включите сред студентите-съавтори на рубриката **не е необходимо** да не сте си взели изпита по „Компютърни архитектури“. Нещо повече, дори да сте получили оценка Отличен (6,00) на изпита все пак имате възможност да дадете и два отговора, достойни за включване в тази рубрика. При подобни случай поради невъзможност за рецидив не възможно да бъдат обявени Вашите авторски права, освен ако авторът лично не се яви и ме помоли за това.

За да се включите в тази рубрика **не е достатъчно** да не си вземете изпита по „Компютърни архитектури“. Редица Ваши колеги дават напълно верни отговори на цял един въпрос, че дори и на 2–3 въпроса. Това за тяхно нещастие не може да им помогне да си вземат изпита защото за оценка Среден (3,00) се изискват верни отговори на поне 6 въпроса.

Във фигурни скоби в началото на всеки отговор е изданието, в което той е присъствал. Освен това за да се привлече вниманието на читателя към величието на даден отговор, част от него е курсивирана с получер вариант на шрифта.

В настоящето издание са включени отговорите, получени на следните изпити:

101. Поправителен изпит за специалност „Математика и информатика“, проведен на 5 април 2013 г. във ФМИ.

102. Поправителен изпит за специалност „Информатика“ – задочно обучение, проведен на 20 юни 2013 г. във ФМИ.

103. Извънреден изпит за специалност „Информатика“ – редовно обучение, проведен на 27 юни 2013 г. във ФМИ.

104. Ликвидационен изпит за всички специалности, проведен на 10 септември 2013 г. във ФМИ.

105. Редовен изпит за магистърски специализации „Софтуерни технологии“ и „Бизнес информатика с английски език“, проведен на 21 декември 2013 г. във ФМИ.

106. Поправителен изпит за магистърски специализации „Софтуерни технологии“ и „Бизнес информатика с английски език“, проведен на 22 февруари 2014 г. във ФМИ.

107. Редовен изпит за специалност „Математика и информатика“, проведен на 20 март 2014 г. във ФМИ.

108. Поправителен изпит за специалност „Математика и информатика“, проведен на 4 април 2014 г. във ФМИ.

109. Първи редовен изпит за специалност „Информатика“ – задочно обучение, проведен на 26 април 2014 г. във ФМИ.

110. Втори редовен изпит за специалност „Информатика“ – задочно обучение, проведен на 3 май 2014 г. във ФМИ.

111. Редовен изпит за специалност „Информатика“, проведен на 23 юни 2014 г. във ФМИ.

112. Поправителен изпит за специалност „Информатика“ – задочно обучение, проведен на 29 юни 2014 г. във ФМИ.

113. Поправителен изпит за специалност „Информатика“ – редовно обучение, проведен на 30 юни 2014 г. във ФМИ.

114. Ликвидационен изпит за всички специалности на ФМИ, проведен на 16 септември 2014 г. във ФМИ.

115. Редовен изпит за магистърски специализации „Софтуерни технологии“ и „Бизнес информатика с английски език“, проведен на 14 декември 2014 г. във ФМИ.

116. Поправителен изпит за магистърски специализации „Софтуерни технологии“ и „Бизнес информатика с английски език“, проведен на 21 декември 2014 г. във ФМИ.

117. Редовен изпит за специалност „Математика и информатика“, проведен на 23 март 2015 г. във ФМИ.

118. Първи редовен изпит за специалност „Информатика“ – задочно обучение, проведен на 16 май 2015 г. във ФМИ.

119. Втори редовен изпит за специалност „Информатика“ – задочно обучение, проведен на 31 май 2015 г. във ФМИ.

120. Поправителен изпит за специалност „Информатика“ – задочно обучение, проведен на 18 юни 2015 г. във ФМИ.

121. Редовен изпит за специалност „Информатика“ – редовно обучение, проведен на 22 юни 2015 г. във ФМИ.

122. Поправителен изпит за специалност „Информатика“ – редовно обучение, проведен на 30 юни 2015 г. във ФМИ.

123. Ликвидационен изпит за всички специалности на ФМИ, проведен на 10 септември 2015 г. във ФМИ.

124. Редовен изпит за специалност „Информатика“ – редовно обучение, проведен на 15 юни 2016 г. във ФМИ.

125. Поправителен изпит за специалност „Информатика“ – редовно обучение, проведен на 27 юни 2016 г. във ФМИ.

126. Ликвидационен изпит за всички специалности на ФМИ, проведен на 10 септември 2016 г. във ФМИ.

Рубриката съдържа и статистическа информация (общо за двата тома) за участващите автори и имената на първенците по активност, определена по схемата гол + пас, т. е. напълно грешен и смешен отговор (гол) носи на автора 2 точки, а частично грешен и все пак смешен (пас) – само 1.

Приятно четене (и учене!) Ви желае вашият преподавател Вл. Сл. Шкуртов.

НАПЪЛНО ГРЕШНИ И СМЕШНИ ОТГОВОРИ

1. 001. Какъв е приносът на Чарлз Бебидж към компютърните системи?

1. 1) {104} Приносът на Чарлз Бебидж към компютърните системи е че той създава първия прототип на цифров компютър.
1. 2) {105} Чарлз Бебидж дори да не е могъл да изработи своят компютър е открил че ще работи с двуична бройна система, както и че ще трябва някакъв вид програма (по-късно измислена от Ада Лъвлейс)
1. 3) {107} Обмисля създаването на диференчна машина, която е завършена през 1822 г. също така обмисля създаването
1. 4) {107} Създава диференчна машина която е създадена 1822. Също така обмисля създаването на аналитична машина
1. 5) {110} Започва създаването на първата аналогова машина

2. 002. Какъв е приносът на Августа Ада към компютърните системи?

2. 1) {105} Създава Програмна Аналитична машина

3. 003В. Защо Огъста Едъ се смята за първия компютърен програмист?

3. 1) {104} Защото написва **аналитична програма**.
3. 2) {105} Помага за реализирането на първата аналогова машина на Чарлс Бейб
3. 3) {105} Огъста Ада осъществява идеята на Бабидж, като написва (създава) първият машинен език за неговата машина
3. 4) {119} Защото е измислил програмируема машина

4. 004. Защо Чарлз Бебидж не може да конструира своята Аналитична машина?

4. 1) {112} Тя трябва да получи алгоритъм и входни данни, след което сама да извърши пресмятанията. Проблемът е как да се оперира с числата и как да се представят. и поради технически причини.
4. 2) {112} идеите му са аналитичната машина да получи алгоритъм и входни данни и след това да извършва пре смятания
4. 3) {115} Идеите му са аналитичната машина да получи алгоритъм и входни данни след което сама да извърши промяната
4. 4) {119} Проблема е как да се представят числата **и как да се оперират**.

5. 007. Какви са трите принципа на Джон фон Нойман?

5. 1) {105} Компютъра е система, която работи в двоична бройна система. Извършва само действие събиране. Ето защо информацията трябва да бъде структурирана по начин разбираем за ЦП.
5. 2) {105} Трите принципа та Джон фон Нойман са: двоична бройна система, запазнате та итфопмацията н памет и
5. 3) {107} 1 принцип – Двуична бройна система (още при АВС)
2 принцип – Програмата да се съхранява в паметта
3 принцип – Достатъчна е само операцията събиране
5. 4) {107} 1. въвеждат се и се извеждат данни от компютъра.
2. управлява и координира връзката между всички компоненти
3. Съхранява информацията – трайна или текуща шина
5. 5) {107} двоична бройна система
програмата да се съхранява в паметта
достатъчна е само операцията събиране
5. 6) {110} – Информацията да се съхранява в паметта на компютъра
– Двоична бройна система още при (АВС) на Джон Атанасов)
– Достатъчно е само действие събиране
5. 7) {114} Важна е операция събиране, двоична бройна система, цели числа
5. 8) {114} Двоична бройна система, програмата да се съхранява в паметта, достатъчна е само операция събиране.

6. 008. Кои два въпроса трябва да получат принципен отговор за да бъдат реализирани идеите на Ч. Бебидж за създаване на компютър – автоматично работеща сметачна машина?

6. 1) {105} Трябва да се избере вид на представяне на числата и да се реши въпроса със временното съхранение на данните.
6. 2) {105} Съхраняване на цифри (данни), чрез релета. въвеждане на данни, изчислямане
6. 3) {105} Вкарване и изкарване на данни.

7. 009. По какъв начин могат да се представят числата в един компютър и как се наричат съответните типове компютри?

7. 1) {115} Цели и дробни числа типжви кжмпютри – I, II, III чип

8. 010. Какъв е принципът на работа на аналоговите компютри?

8. 1) {105} Операциите при тях се изпълняват с електронни схеми, чиито работни характеристики моделират процеса
8. 2) {105} Принципът на работа на аналоговите компютри е че те са свързани с диференциални уравнения по трудни по вход и изход
8. 3) {106} Принципът на работа на аналоговите компютри е сравнително прост. При тех изходното напрежение на една схема е равно сбора от напрежението на други две входни схеми
8. 4) {115} Данните в аналоговите компютри не се предават по цифров път

9. 012. Какъв е принципът на работа на цифровите компютри?

9. 1) {105} Цифровите компютри се наричат още компютри с дискретно действие. При тях се оперира с данните като резултата, за разлика от аналоговите където е изходно напрежение, сума на две входящи, при цифравите е резултат които отговаря на условията Да, не, И, Или, Не-Или, Да-или …
9. 2) {105} Машини с прекъсваемо действие
9. 3) {115} Числата се представят като елементи с n-устойчиви състояния в р-ична ПБС
9. 4) {115} Принципът на работа на цифровите компютри е чрез работа с числа, наричан се и дигитални. **[Българино! Знай своя род и език!]** Аналоговите са с непрекъснато действие. Цифровите **работят с позиционни бройни с-ми**. Прекъснато действие

10. 013. Как още се наричат цифровите компютри?

10. 1) {110} Компютри от 3-то поколение (енергозависими компютри)

11. 015В. Посочете поне два недостатъка на аналоговите компютри.

11. 1) {118} Аналоговите компютри са по.бавни, заемат повече пространство, по-трудно се осъществява вход и изход

12. 016А. Посочете поне две предимства на цифровите компютри.

12. 1) {105} По-бързи; по-евтини

13. 016В. Посочете поне два недостатъка на цифровите компютри.

13. 1) {105} По-малка точност на изчисление
Двоична система (сложна за човека
Общо предназначение ⇔ тясно специализиране
13. 2) {115} – ~~Възможни грешки при входа и изхода~~
– Смятат бързо но са възможни грешки

14. 018. По какви критерии (поне 2) могат да бъдат класифицирани компютрите?

14. 1) {109} Те могат да бъдат класифицирани по бълзодействие или по обем памет. Бързодействието зависи от Ц.П., а обем памет от ОП в която се зареждат програмите.
14. 2) {112} Могат да се класифицират по бързодействие на ЦП или по обем на ОП.
14. 3) {114} Специални и с общо предназначение **[Това наистина са страхотни критерии!]**
14. 4) {116} Цифрови и аналогови

15. 021. Как по друг начин бихте нарекли компютрите със специално предназначение?

15. 1) {114} Компютрите със специално предназначение се наричат **суперкомпютри** Те могат да работят с една специална предназначена програма например контролерите.

16. 022. Какви са характерните особености на компютрите с общо предназначение?

16. 1) {115} Компютрите с общо предназначение са универсални компютри.

17. 023. Как по друг начин бихте нарекли компютрите с общо предназначение?

17. 1) {110} Intel
17. 2) {112, 114} персонални компютри

18. 024. Кой вид компютри съответстват по-пълно на идеите на Ч. Бебидж – универсалните или специализираните? Защо?

18. 1) {107, 107} Специализираните
18. 2) {112} Чарлз Бебич е създател на сметачна машина, чието предназначени съответства на специализираните компютри.
18. 3) {112} специализираните

19. 026. По какъв критерий компютрите се делят на поколения?

19. 1) {110} По устройството (изчислителна мощност) на ЦП и мащабите на компютрите
19. 2) {115} Днес се използват компютри от 4-то поколение, различаваме ги по вида на елементите /лампови, транзисторни/
19. 3) {123} По това дали са аналогови, цифрови или хибридни.

20. 027. Каква е елементната база на компютрите от първо поколение?

20. 1) {105} Елементната база на компютри от първо поколение е електрически релета
20. 2) {105} Елементната база на компютрите от първо поколение са релета
20. 3) {106} Изграждат се на базата на релетата
20. 4) {106} Релета

21. 028. Каква е елементната база на компютрите от второ поколение?

21. 1) {104} Елементната база на компютрите от второ поколение е **централен процесор, екран**
21. 2) {105} Компютрите от II покобевие са с елементна база ел. лампи
21. 3) {106} електр. лампи
21. 4) {110} **Релета**, транзистори.

22. 029. Каква е елементната база на компютрите от трето поколение?

22. 1) {105} При компютрите от трето поколение се използват кондензатори
22. 2) {105} Елементната база на компютри от третото поколение са транзистори. Първо поколение е реализирано с релета. Второ поколение с електронни лачпи
22. 3) {106} Елементната база на компютри от третото поколение е транзистори.
22. 4) {106} Елементната база на компютрите от третото поколение са транзисторите (в дискретен вид).

23. 030. Към кое поколение принадлежат съвременните компютри и какво е тяхната елементна база?

23. 1) {119} Трето поколение. С 64 bit елементна база.

24. 031. Какви тенденции се наблюдават с промяната на конструктивните елементи на компютрите (поне 3)?

24. 1) {115} Тенденции към: 1. Намаляване на размера 2. Намаляване потреблението на електричество 3. Намаляване на цената за памет на всеки 3 години пада наполовина

25. 041. Какво представляват адресите и каква е тяхната роля в електронните системи?

25. 1) {102} Идентифицират къде е записан обекта в ОП.
25. 2) {102} Достъп до регистър на ЦП. Винаги се използват акумолатори, адресните регистри са ограничени.
25. 3) {117} Адресите са указатели към част от паметта – те показват къде точно в паметта трябва да се запише дадена информация

26. 043. Какво ограничава размерът на адресната шина?

26. 1) {102} Размера на адресната шина се определя чрез големината и броя на шината.

27. 048. Какво е предназначението на ОП?

27. 1) {104} **Изпълнява** изментните операции зададени от потребителя.
27. 2) {104} Предназначението на ОП е да съхранява основната информация на един компютър.
27. 3) {110} В нея се съхраняват операционната с-ма и изпълнимите програми, поради което тя се намира в непрестанно взаимодействие с процесора и външните устройства. От нея в голяма степен зависи производителността на компютъра като цяло.
27. 4) {110} ОП е предназначена за моментно съхранение на данни
27. 5) {113} **Обработване на данните** изпратени от ЦП към ОП.
27. 6) {113} Предназначението на ОП е да **съхранява временни файлове и програми**
27. 7) {114} **Изпълнява** изментните операции зададени от потребителя. Предназначението на ОП е да съхранява основната информация на един компютър
27. 8) {115} Място за съхранение на входни и изходни данни, също и **част от кода на програмата**, която се изпълнява
27. 9) {121} Да съхранява в себе си информацията от програмите.
27. 10) {123} Съхранява входни, изходни и междинни данни.
27. 11) {124} Целта на ОП е да да подпомага за изпазването на информацията
27. 12) {124} Предназначението на оперативната памет е процесорът да борави с нея.

28. 049. От какво е изградена (се състои) ОП? Защо?

28. 1) {104} Оперативната памет (ОП) бива два вида RAM и ROM енергозависима и енерго–независима
28. 2) {107} ОП е изградена (състои) се **от запомнящи устройства** с две устойчиви състояния (0 и 1), наречени битове.
28. 3) {110} От клетки (т. нар. „адреси“), които позволяват организиране на информацията.
28. 4) {110} Оперативната памет е бърза флаш памет която носи основна информация за компонентите и не се нулира.
28. 5) {111} представлява все едно едномерен масив с n на брой клетки които имат запомнящ се елемент. Всеки елемент в ОП си има адрес и е с размер 1 bit
28. 6) {113} адресация, индексиране, запис
28. 7) {114} съвкупност от записваеми елементи с две устойчиви състояния
28. 8) {115} ОП е изграден от множество абсолютно еднакви клетки памет (2^K), които се различават единствено по адреса си. Така се реализира бързодействие за едно и също време може да се достъпи коя да е клетка от паметта
28. 9) {115} ОП се състои от клетки обединени в байтове
28. 10) {116} Оперативната памет е изградена **от клетки обединени в битове**.
28. 11) {121} Оперативната памет се състои от блокове които проверяват определената операция и ако отговаря на изискванията я препраща към ЦП за изпълнение.
28. 12) {124} клетки
28. 13) {124} Клетки, които се състоят от n на брой компоненти, наречени битове, като 8 бита = 1 байт. Клетките на ОП са еднакви, но с различна адресация Битовете сформират клетки чрез адресното пространство.
28. 14) {126} От клетки, всяка от които се състои от 8 бита.

29. 050. Какво представлява понятието „клетка от ОП“?

29. 1) {102} Понятието клетка с ОП представлява място в което се съхраняват данни за изпълнението на даден процес.
29. 2) {102} Клетка памет е физическа памет в която се съдържа осем Бита информация. Част е от RAM паметта и има адрес.

29. 3) {102} Клетка = дума, място в паметта за съхрна, Всене на информация с големина равна на големината на думата. Различните производител дефинират и различна големина на думата.
29. 4) {105} Всяка клетка е номерирана, има точно определен адрес, клетките съхраняват информацията. n = 8 и клетките се наричат байтове
29. 5) {106} Клетка от ОП обединява „n“ (определен брой) бита като в съвр. компютри 1 клетка = 8 бита = 1 байт
29. 6) {112} **Елементарно устройство**, състоящо се от един транзистор и един кондензатор и което може да съхранява данни.
29. 7) {113} Когато с два значения разряда
29. 8) {114} всички битове се групират по „N“ на брой в клетка
29. 9) {114} Всички битове се групират по n броя и една група се нарича клетка
29. 10) {115} Клетката от ОП представлява група n от записаеми елементи с 2 устойчиви състояния имаща пореден номер и адрес.
29. 11) {115} ОП – оперативна памет – съхранява входни и изходни данни, междинни резултати и кода на програмата.
29. 12) {119} Клетката от ОП – може да чете и да записва и винаги чете последователно
29. 13) {119} Клетката най малката група в ОП.
29. 14) {120} Клетката съдържа битове памет записани като двоичен код. Нейната големина е една и съща, като единствената разлика е в нейната адресна част.
29. 15) {121} Група от битове
29. 16) {121} Всички битове се групират по n броя и една такава група се нарича клетка
29. 17) {122} – клетка от ОП наричаме мястото където съхраняваме данните. Всяка клетка съдържа в себе си 8 бита, те се равняват на 1 байт.
29. 18) {122} Това е косвена регистрова дейност
29. 19) {123} **Клетка** от 8 бита памет
29. 20) {124} Клетките на ОП са с еднакъв размер, но различни адреси. Размерът се определя от проектанта
29. 21) {125} Групиране битове, защото един бит съдържа твърде малко информация
29. 22) {126} Клетката в ОП е единицата в която се съхраняват данните заредени в нея. Ако размера на клетката е по-малък

30. 051. Защо битовете на ОП се обединяват в клетки?

30. 1) {104} Битовете на ОП се обединяват в клетки за да бъдат адресирани.
30. 2) {105} Битовете на ОП се обединяват в к-клетки, защото една клетка може да съдържа само 8 бита.
30. 3) {109} Защото когато се обединят в клетки те имат адрес с който биват използвани по лесно от ОП.
30. 4) {111} Заради адресното пространство, за да може с адрес с по-малък размер да се адресират данни. Също така и от технологична гледна точка.
30. 5) {111} За бързо и оптимално намиране на данни
30. 6) {114} Заради адресното пространство, за да може с адрес с по-малък размер да се адресират данни. Също така от технологична гледна точка
30. 7) {116} Заради адресните пространства за да може с адрес по-малък да се адресира данни. Също така от технологична гледна точка
30. 8) {121} Защото самата ОП е конструирана по този начин
30. 9) {123} Защото 1 байт заема 1 клетка от паметта.
30. 10) {123} Защото по отделно са прекалено малки. Обединявайки се в клетки те може да се обработват заедно
30. 11) {124} Заради адресното пространство, за да може с адрес с по-малък размер да се адресират данни. Също така и от технологична гледна точка
30. 12) {124} По лесно търсене, като намерим първия бит на клетката.
30. 13) {126} по този начин ги различаваме. Клетките са номерирани.

31. 052. Как може да се осигури достъп до единичен бит в клетка от ОП?

31. 1) {124} битовете в клетката се номерират трябва да се знае адреса на клетката
31. 2) {124, 125} Битовете могат да се номерират от n–1 или от n до 1
31. 3) {125} Като знаем адреса ѝ я сростъпваме директно.

32. 054. От какво се ръководят конструкторите, когато определят размера на клетката в ОП на даден компютър?

32. 1) {104} От броя на битовете в нея.
32. 2) {105} От размера на параметрите, които са зададени в програмата
32. 3) {105} Конструкторите се ръководят от това какви типове методи се въвеждат, за да може да се определи размера на заделените клетки в ОП
32. 4) {113} От вида на паметта, предназначението на компютъра
32. 5) {115} От големината на числото, което трябва да съхранят
32. 6) {116} Проектантът взема предвид стойността на данните, с които той **[Т. е. проектантът!]** ще работи и принципът за обратна съвместимост
32. 7) {121} **Размерът на една клетка е един бит** В една клетка може да бъде записан само един бит информация, н. е. една „1“ или една „0“.
32. 8) {122} Битовете в ОП се групират по (n) броя в клетка

33. 055. Какво представлява ОП от логическо гледище?

33. 1) {104} Предава алгоритъма към изчеслилителното у-во.
33. 2) {104} Оп е изградена от запомнящи елементи с две устойчиви състояния (0 и 1) наричани двоични разреди или битове
33. 3) {105} ОП е последователност от клетки започващи от 0000 до FFFF, клетките са в основата на думи. 8 bits = 1 byte – дума.
33. 4) {105} ОП от логическо равнище би трябвало да представлява тази част от ОП, която е отговорна за осъществяването на действията на АЛУ на ЦП.
33. 5) {107} Оперативната памет представлява място за съхранение междинни данни и съхранение на изпълняваната в момента програма
33. 6) {110} Стек с машинни инструкции, обработващи се по приоритет.
33. 7) {110} Това е паметта (даните) с които оперира процесора. Тя се зарежда в RAM паметта и седи на изчакване докато не притрябва на процесора.
33. 8) {110} От логическа гледна точка ОП може да се разглежда като стек.
33. 9) {111} ОП е изградена от запомнящи елементи с 2 устойчиви състояния 0 и 1, наречени двоични разряди или битове
33. 10) {111} ОП е изградена от запомнящи елементи с две устойчиви състояния, наречени двоични разряди или битове. Понеже един бит съхранява твърде малко информация, битовете се групират в клетки.
33. 11) {111} ОП представлява място за съхранение на междинни данни от програма, по време на нейното изпълнение.
33. 12) {111} Оперативната Памет е изградена от запомнящи елементи с две състояния 0 и 1 наречени битове. Понеже един бит съхранява твърде малко информация, битовете се групират по n броя. Групата от битове се нарича клетка. Клетките се различават като се номерират с числа наричани адреси.
33. 13) {113} ОП – е памет в която се съдържа изчислението да момента когат те вече няма да трябват
33. 14) {114} ОП е съвкупност от байтове (клетки) Там се запаметяват (запомнят) МИ
33. 15) {121} ОП е временно хранилище на данни
33. 16) {122} ОП е изградена от запомнящи елементи с две устойчиви състояния 0 и 1, наречени двоични разряди или битове. Понеже един бит съхранява твърде малко информация. Битовете се групират по n броя и една такава и една такава група се нарича клетка. Клетките се различават като се номерират с числа от 1 до $2^{к-1}$ наречени адреси. Битовете в клетките се номерират от n–1 до нула или n до 1

33. 17) {122} Склад за временно съхранение на данни.
33. 18) {123} Оп е изградено от запомнящи елементи с две устойчиви, които се наричат двоични разреди или битове Понеже един бит съхранява твърде малко информация битовете се групират по n броя и една такава **клетка се нарича клетка** клетките се номерират
33. 19) {124} ОП се използва за извършване на операция в различни устройства
33. 20) {124} Представлява памет (хранилище) с което си борави процесора
33. 21) {125} ОП е изградена от запомнящи елементи с две устойчиви състояния (0 и 1), наречени двоични разряди и битове.
{125} Оперативната памет представлява място за съхранение на алгоритми в програмата и кода, който е оперативен в маметта

056. Какъв е размерът на клетките на съвременните ОП и как се наричат те?

33. 22) {102} Един компютър има битове в клетките (**битови клетки**) Размерът на клетките **е битов**, колкото повече бита има в една клетка тя е толкоз голяма
33. 23) {112} Съвремените ОП използват 8 клетки, които се наричат – 8 бита или 1 байт.
33. 24) {113} Клетките се наричат битове, а размера на една клетка е 8 бита
33. 25) {123} **Размера е** за слетка **1 бит**. Наричат се **битови клетки**.

34. 057. Какви са характерните черти (поне 3) на ОП?

34. 1) {103} Основни характерни черти на ОП
– не работи когато компютърът е изкл.
– честотата и се измерва в мегохерцове
– има различни видове ОП като DDR, DDRII,
34. 2) {105} **Оперативната памет е памет** в която се съхранява програмата **в която се изпълнява** както и текущите данни
34. 3) {121} Бърз достъп, кратко време за реакция, може да се използва многократно без забавяне. Всеки 1 год.се намаляват изискванията а тя се увелич.
34. 4) {121} **За** съхранение на данни, **обработка на данните**, извличане на данни,
34. 5) {122} Оперативната памет е бърза, енергозависима и евтина.
34. 6) {124} Енерго зависима **[И магнитната ли?]**, бърза и скъпа за производство **[И когато си използват интегрални схеми ли?]**
34. 7) {125} Съхранява данни, енергозависима, .

35. 058. Каква е главната характерна черта на ОП?

35. 1) {101, 103, 123} Висока скорост на обмен на данни
35. 2) {104} Времето за достъп до произволна клетка от паметта, което е пъти по-голямо от времето за четене/запис на информацията в клетка.
35. 3) {104} Главната характерна черта на ОП е, че първо запомня информацията, а след това я прудявя
35. 4) {114} Да запаметяват и изпълняват.
35. 5) {116} ОП – Оперативна памет – запомня данните. **Съставена** е **от клетки с 2 устойчиви състояния**
35. 6) {119} ОП – оперативната памет (ROM) – тя е енергозависима. Тя служи за съхраняване на междинни данни.
35. 7) {121} Цената на ОП намалява наполовина приблизително на всеки 3 години.
35. 8) {122} Времето за достъп до произволна клетка
35. 9) {122} Главната черта е, че е енергозависима **[И магнитната ли?]**
35. 10) {123} Служи за запазване на данните докато програмата се изпълнява.
35. 11) {124} МП не се записва право в ОП. След първият пас става ясен размерът на програмата в ОП и ОП се моделира във ВП.
35. 12) {125} Висока скорост на четене и запис.
35. 13) {125} Разделена е на адресирани клетки от n бита (8) и може да симулира асоциативна памет ако е нужно.

35. 14) {126} Времето за достъп до произволна клетка от паметта (seek time) В пъти е по-голямо от времето за четене/запис на информацията в клетка.

36. 060. Какви видове памет съществуват по отношение на механизма на осъществяване на операциите на паметта?

36. 1) {105} Динамична и статична
36. 2) {115} Памет за адреси и памет за данни
36. 3) {119} RAM (Random Access Memory) и ROM (Read-Only Memory) RAM – памет с произволен достъп, позволява многократно четене и записване на данни. Недостатък – информацията се губи при изключване на системата, не може да се използва за дълготраен запис на данни. ROM – памет само за четене; съдържанието се записва от производителя на дадената машина и не може да се променя след това, освен ако не се замени паметта с друга. Използва се за дълготраен запис на данни, които се използват многократно, без да се променят. Например за съхранение на BIOS.

37. 061. Какъв вид памет е ОП на съвременните компютри: адресна или асоциативна? Защо?

37. 1) {105} Оперативна памет служи за съхраняване на изпълняваната в момента
37. 2) {108} Асоциативна.
37. 3) {111} По-удобно е да бъде асоциативна
37. 4) {123} Асоциативна кеш памет.
37. 5) {123} ОП – оперативната памет е асоциативна тъй като е по-бърза от адресната

38. 064. Как физически се изгражда ОП и защо?

38. 1) {104} ОП е изградена от клетки които съдържат n на брой битове. В днешните компютри n=8. Всички клетки са еднакви с изключение на техния адрес. ОП също може да се представи като едномерен масив. ОП е енергозависима. При пускането на компютъра тя е празна. Има и енергонезависими памети които не може да бъде променяна. Такъв пример е BIOS, където се съдържа програмата за проверка на компютърната система. В ОП адресите на клетки са цели числа.
38. 2) {104} физически ОП се изгражда от битове 0 и 1, които се разполагат по 8 в **клетка, която се нарича адрес**.
38. 3) {105} Оперативната памет има адрес и стойност. На определен адрес отговаря определена стойност. Тя се изгражда от стека и хайпа. В стека се намират адресите на инструкците които се изпълняват.Стека е подредени с дадена последователност и изпълняви и зарежда инструкците. В хайпа данните са разоврани.
38. 4) {105} ОП физически е разделена на клетки bits 6 bits = 1 byte
38. 5) {105} Електроника, енергозависима, при изключване ОП губи съдържанието си. за сметка на това е бърза.
38. 6) {105} ОП е разделена на определен брой клетки, всяка една от които с различен АП за по-бързо и по-лесно разпространение и съхраняване на данни.
38. 7) {106} Изгражда се, чрез набирането на информация и действието на механизмите свързващи я към централния процесор. Защото така се улеснява действието ѝ и скоростта ѝ.
38. 8) {110} Батовете се номерират и поставят под стекове.
38. 9) {110} Оперативната памет е за запомняне на данните. Съхранява входни/изходни данни, межд. резултати и кадъ е програмата. Адресна и асоциативна бива ОП. Физически се изгражда с две уст. състояния 0 и 1.
38. 10) {113} ОП е изградена от клетки с различни адреси и n на брой битове. В днешните компютри n = 8 По този начин адресите са по-къси тъй като няма да се адресира всеки бит.
Използва се DRAM, защото:
– е по-евтина
– по-голяма плътност за единица място.

38. 11) {116} Физически се изграждат памети с две устойчиви състояния, съответстващи на 0 и 1. Защото този вариант е най-близък до двоичната ПБС, която се използва в компютрите. Освен това полупроводниковите ИС са евтини и с възможвост за голяма степен на интеграция.
38. 12) {122} Физически се изграждат **памети с две устойчиви състояния 0 и 1**. Освен това полупроводниковите ИД са евтини с възможност за голяма степен на интеграция
38. 13) {123} ОП или първичната памет се изтражда он двоични разреди наречени битове.
38. 14) {124} Брояч за кратност на клетки, физицен блок за по-голяма скорост при пренос
38. 15) {125} Физически се изгражда памета с две устойчиви състояния съответстващи на 0 и 1. Защото този вариант е най-близък до двоичната ПБС, Която се използва в компютрите
38. 16) {125} ОП е изградена от едномерен масив от клетки, в които клетки се съхранява информация 1 клетка = 1 byte = 8 bit

39. 065. Какви технологии могат да се използват за направа на ОП?

39. 1) {105} Чрез транзистори или чрез ИС с кондензатори
39. 2) {111} **механично**, машинно
39. 3) {121} Шини за вход/изход на данни, механизъм за запомняне на съответните данни

40. 066. Какви са предимствата и недостатъците на електрическите памети?

40. 1) {114} Предимствата и недостатъците на електрическите памети.
Предимствата са че електрическите памети съхраняват напълно информацията в **пъин** размер.
Недостатъците че при внезапно прекъсване на ел. захранването **съществува** вероятност от загуба на данни.
40. 2) {114} бързи, лесни за изработване при интегрални схеми и компактни **[Не е ясно дали авторът на този зашеметяващ отговор смята описаното за предимство или за недостатък!]**
40. 3) {117} С цел да се скъсява АП
40. 4) {121} Електрическите памети се поддържат само на ел.ток,не се поддържат на батерии Електрическите памети са по-евтини.
40. 5) {124} Недостатъци: по-стари, по-бавни; Предимства: по-евтини

41. 067. Какво е следствието от използване на електрически принципи за запомняне при паметите?

41. 1) {103} Необходим е ток. Изключването анулира съдържанието.
41. 2) {103} Създаване на електрически пакети които могат да бъдат енергозависими в зависимост от технологията.

42. 068. Как влияе енергийната зависимост на електрическите памети върху изграждането на ОП? Защо?

42. 1) {116} Енергийната зависимост е един от основните недостатъци на електрическите памети
42. 2) {118} Електрическите памети са напълно зависими от ел. ток, при изключване на компютъра всичко записано върху ОП се губи.

43. 069. Какви интегрални схеми постоянна памет (ROM) познавате (поне 3) и по какво се различават те?

43. 1) {114} ROM не е енергозанисима за разлика от RAM
ROM-а съхранява данните дори и след изкл. на компютъра.
43. 2) {115} статична, динамична, енергозависима

44. 070. Какви интегрални схеми изменяема памет (RAM) познавате и по какво се различават те?

44. 1) {125} Маскова постоянна памет, програмируема, постоянна, изтриваема програмируема постоянна памет, изменяема програмируема

44. 2) {126} Статична SRAM и статична DRAM различават се по елементната база е цена.

45. 071С. Посочете поне две предимства на схемите статична памет в сравнение със схемите динамична памет.

45. 1) {105} минимална загуба на данни

46. 072. Кои видове интегрални схеми изменяема памет (RAM) – статична или динамична, са предпочитани в съвременните компютри и защо?

46. 1) {111} Статична (SRAM) и динамична (ДRAM) Различават се по–елементна база вSRAM се състои от транзистори, а в DRAM от кондензатори

46. 2) {124} В съвременните компютри се използват интегрални схеми изменяема памет. **[Статична или динамична?]**

47. 074. Какво представлява понятието „дума“?

47. 1) {104} n на брой битове данни. В различните компютри размера на думата е различен. Интер: 2 бита, IBM – 4 бита.

47. 2) {104} Понятието „дума“ представлява израз чрез който се указва дадена мисъл с нея се записват и възпроизвеждат мисли и др.

47. 3) {105} Понятието „дума“ е поредица от последователни битове

47. 4) {105} Думата е 8 битова, има и двойна „дума“ 16 бита

47. 5) {113} **Число**, което е цяло и без добавени знаци

47. 6) {114} информация записана в байтове

47. 7) {119} Понятието дума представлява, основна структурна **езикова** единица.

47. 8) {122} Група от битове, които централния процесор обработва

47. 9) {122} Поредица от последователно подредени битове

47. 10) {124} Това е редица от символи, с определено значение.

47. 11) {124} Обединение от клетки

47. 12) {124} Съвкупност от клетки, в които има битове. N=8

48. 075. Какво представлява понятието „размер на дума“?

48. 1) {102} Понятието размер на дума се определя от броя на байтовете в които е разположена тя. – Кратен е на клетката.

49. 076. Каква е връзката между размера на думата и броя на битовете в клетките на един компютър? Защо?

49. 1) {102} Размера на думата се разбива по букви като на всяка буква отговарят 4 символа от #, докато битовете в една клетка побират 8 (примерно) символа, тоест всяка клетка след адресната събира максимум от 2 символа но не е задължително. Руски системи работят на 32, 64, 86 или 128 bit и това определя отношението им спрямо данните

49. 2) {102} Размеря на шината се определя от броя на паралелните процеси които може да извърши централният процесор (ЦП

49. 3) {102} Колкото **по –големи са битовете** на клетките, толкоз по голям ще е размера на думата.

49. 4) {103} Битовете определящи размера на думата са **крайни** на битовете на клетката

49. 5) {104} Връзката е, че един символ заема един бит памет.

49. 6) {105} Всяка дума се дули на различен брои битове за да може да я разбере компютърът (превежда се на компютърен език) а битовете се съхраняват в клетки. Колкото повече битове толкова повече клетки

49. 7) {105} Връзката между размера на думата и броя на битовете е, че с увеличаване на размера на думата се увеличават и броя на битовете. Това се получава, защото една клетка има определен размер.
49. 8) {106} Колкото **по-голяма е думата по-големи са и битовете**, защото всеки знак се разчита и с увеличаване на знаците се увеличават и битовете.
49. 9) {109} Колкото **по-голяма е думата**, толкова по голям брой битове са нужни.
49. 10) {111} Броя на битовете в клетките на един компютър е кратен на размера на думата, защото клетките са съставени от думи.
49. 11) {114} И при двата варянта се използват битове в клетки.
49. 12) {115} Размерът на думата е краен на степен на 2-ката брой клетки на ОП.
49. 13) {121} Всяка клетка е един бит.
49. 14) {121} Размера на думата определя какъв е броя на битовете. Всяка една буква е с определен размер бит. Колкото по-голяма е думата толкова по-голямо място заема в клетката. Всяка една клетка има определен размер.
49. 15) {122} В зависимост от размера на думата се определя брой на битовете в клетката и обратното. Като всеки бит определя буква от азбуката за да се формира дума.
49. 16) {122} Компютърът може да разчита думи не по-големи от брой на битовете му
49. 17) {123} Размер на думата представлява броя на битовете, които са използвани за образуването ѝ.
49. 18) {123} Една дума заема 8 поредни клетки (8 бита) които е равно на 1 байт.
49. 19) {124} Размерът на думата е = 8 bit 1 byte = 1 клетка Една дума е в една клетка.
49. 20) {126} Поредица от битове се нарича байт, поредица от байтове – дума. Размера на думата определя дали дадно дума може да се зареди изцяло в дадена клетка.

50. 077. Как се постъпва, когато битовете на една клетка от ОП не са достатъчни за запис на данните от даден тип?

50. 1) {102} Левите старши битове с по-малък адрес при IBM
Левите старши битове с по-голям адрес при Intel
50. 2) {102} Левите битове на думата в клетката с по-голям адрес – Intel, DEC
50. 3) {115} Когато битовете на една клетка от ОП не са доститъчни за запис от даден тип се преминава към друга клетка.
50. 4) {115} Битовете могат да съхраняват твърде малко информация. Поради това те се обединяват в клетки.

51. 078. Как в ОП се идентифицират думи, разположени в повече от една клетка? Защо?

51. 1) {103} Когато по време на изпълнение програмата може да бъде преместена
51. 2) {110} Чрез дублиране на клетки
51. 3) {110} Когато думата не се събира в една клетка, то тя се дописва в следващата най-близка (по адрес) клетка.
51. 4) {114} Всяка дума има адрес
51. 5) {115} Разполагат се последователно, защото като знаем адреса на клетката е по-лесно да ги достигнем.
51. 6) {125} Идентифицират се с адреса на първия байт от думата или посредник.

52. 079. Какво представлява понятието „интегрални граници“?

52. 1) {103} електронна схема с миниатюрни размери, състояща се от полупроводникови устройства и пасивни компоненти

53. 080. Защо производителите изискват разполагане на думи в ОП от интегрални граници?

53. 1) {103} За да се избегне препълване на ОП
53. 2) {104} Защото е по-евтино.
53. 3) {105, 122} За да не се получава застъпване на клетки.

53. 4) {106} За да няма смесване.
53. 5) {111} За по-голяма ефективност на работата с оперативната памет. (ОП).
53. 6) {115} За по-голяма ефективност на работа с Оперативната памет
53. 7) {116} За да не се получава застъпване на клетки
53. 8) {119} Проблемът се състои в това че разположението може да стане по два начина – старшите битове могат бъдат в клетката с по малък адрес или с по голям адрес
53. 9) {119} За да не се получава запълване на клетката.
53. 10) {121} За да не се получи „застъпване“ на клетки
53. 11) {121} Производителите изискват разполагане на думи в ОП, защото така скоростта за изпълнение на задачата е по-голяма.
53. 12) {121} Тъй като производителите могат подробно да разгледат съвместимостта същевременно да изберат най-доброто от което имат нужда за да избегнат грешки, затруднения в по-късен етап.
53. 13) {121} Защото трябва да се избегне препълване.
53. 14) {122} За да не се получава „застъпване на клетки“
53. 15) {122} Централният процесор се състои от шина, устройство за вход/изход, конструктор.

54. 081. Какви проблеми създава записът на една дума данни в две клетки от паметта?

54. 1) {102} Трябва да се помни адресът на втората клетка.
54. 2) {104} Забавя процесора
54. 3) {104} Проблемът е че системата се натоварва по този начин и може да се наложи обработката на два пъти повече данни и да зацикли, или забие
54. 4) {104} Записа на дума в две клетки създава проблем с адресирането, тъй като не се заделят две последователни клетки, а две последователни празни клетки, което предполага намеива форма на адресация от едната към другата клетка.
54. 5) {105} Когато думата е прекалено голяма и информацията не може да се събере само в една клетка от паметта тя се записва в друга. Проблемите, които могат да възникнат са по бавно търсене и адресиране на информацията.
54. 6) {115} маломерни къси голямомерни и обратно
54. 7) {115} Записът на „дума“ в две клетки създава проблеми при прехвърлянето на данни от една ОП в друга. Може да се получи проблем със записа, и по-скоро с малокрайните и голямокрайни адреси.

55. 082A. Как се решават проблемите по запис на една дума в две клетки с последователни адреси?

55. 1) {109} Чрез адресации
55. 2) {111} Старшите битове се записват в клетка с по-малък адрес (**голямо кратен компютър**) или в клетката с по-голям адрес
55. 3) {113} чрез промяна на оператора за индексиране на данните
55. 4) {114} При разполагането на дума в клетки с по малак адрес или по голям
55. 5) {126} При преноса на данни (> 1 байтове) и две различни машини (в организацията за съхранение на данни които са по големи от една дума)

56. 082B. Кога се проявяват проблемите по запис на една дума в две клетки с последователни адреси?

56. 1) {105} Когато размера на една дума изисква запис в повече от една клетки от паметта (обикновено две) производителите решават дали левите (старши) битове на думата да бъдат записани в клетките с по-голям или по-малък адрес. От това могат да се породят проблеми свързани с начина на запис, а в последствие и четенето.
56. 2) {105} При преместване на програмата към друг ЦП
56. 3) {105} Проявява се при запис на думата във втората клетка
56. 4) {106} Когато се извлича записа от клетките

56. 5) {111, 111} **Проблемът се състои в това**, че разполагането може да стане по 2 начина – старшите битове на думата могат да бъдат в клетките с по-малък адрес или с по-голям адрес.
56. 6) {115} Когато имаме пренос
56. 7) {116} Проблемът се състои в това, (е разполагането може да стане по 2 начина: старшите битове могат да бъдат в клетките с по-малък адрес (голямокраен компютър) или в клетките с по-голям адрес (малокраен компютър)
56. 8) {123} Когато клетките не са свързани една с друга
56. 9) {123} Проблемите по запис на дума (поредица от битове) в 2 клетки от паметта с последователни адреси, се проявяват когато се определя от коя клетка да се прочете адреса на МИ.
56. 10) {125} Проблемът се състои в това, че разполагането може да стане по два начина – старшите битове на думата могат да бъдат в клетката с по-малък адрес (голямокраен компютър) или с по-голям адрес (малокраен компютър)

57. 083. Какви типове цели числа могат се представят в съвременните компютри?

57. 1) {102} Цели положителни и цели отрицателни
57. 2) {111, 113, 114, 122} Естествени, **рационални**, цели.
57. 3) {115} byte, int, short, long
57. 4) {115} Всички **реални** числа
57. 5) {122, 124, 126} естествени, цели и **рационални**.
57. 6) {125} **естествени**,
57. 7) {126} С положителен и **утрицателен** знак
57. 8) {126} Положителни и отрицателни.

58. 085. Как се кодира стойността на целите без знак при компютрите?

58. 1) {105} Кодира се с прав код.
58. 2) {105} с Прав код
58. 3) {110} 0 ÷ 0,8 V = 0, 2 ÷ 5 V = 1
58. 4) {110} Като добавим един допълнителен бит, **който представя знака**.
58. 5) {112} bn –1 * 2n – 1 + bn –2 * 2n – 2 + … b1 * n1 + b0*20
58. 6) {112} Като се превръща в двоичен код.
58. 7) {114} Диапазон (0, 2 – 1) (–2n – 1, 0)
58. 8) {115} Като Абсолютна
58. 9) {117} Ако битовете са b_{n-1}, b_{n-2}, … b_1, b_0, то думата е стойността на двоичното число b_{n-1}, b_{n-2}, … , b_1, b_0 ($b_{n-1}.2_{n-1} + b_{n-2}.2_{n-2} + … + b_{1.n1} + b_{0.20}$).
58. 10) {117} Към всички числа от интервала [–2n–1, 2n–1–1] се добавя 2n–1 за преход към [0; 2n–1]

59. 086. Какъв е диапазонът на представимите с n бита цели без знак?

59. 1) {105} 0 до
59. 2) {105, 111} 2^n–1
59. 3) {110} 2n–1–1 до 2n–1–1
59. 4) {111, 113} Диапазон от (2n–1) до 0 или от 0 до 2n–1
59. 5) {114} 2n
59. 6) {121} N^{-1} **[Изключителен диапазон!?!]**
59. 7) {123} Диапазона е **неопределен**.
59. 8) {124} [0; 2n–1] или [–2n–1; 0]
59. 9) {124} от 16 до 64
59. 10) {125} от n^2 до n^{-2}
59. 11) {125} [0, 2^{n-1}–1] или [–(2^n–1), 0]

60. 087. Какви варианти за кодиране на цели със знак познавате?

60. 1) {107, 107, 109, 114} Възможни са две решения
1) абсолютна стойност и знаците „+“ и „–“
2) допълване на отрицателните числа с предварително определено число

60. 2) {105} 1) абсолютна стойност и знаците „+“ и „–“
2) допълване на отрицателните числа с предварително определено число

60. 3) {112} Набор от инструкции единици и нули.

60. 4) {113} Възможни са две решения ① – абсолютна n стойност и знаците + –; ② допълване на отрицателните чесла с предварително определено число

60. 5) {114} Има две решения: първото е абсолютна стойност и знаците + и –
второто е: допълване на отрицателните чесла с предварително определено число
[Вариантите си имат имена, които студентът трябва да знае!]

60. 6) {121} абсолютна стойност; допълване на отрицанелно числа

60. 7) {122} Възможни са две решения:
1. – абсолютна стойност и знаците „+“ и „–“
2. – допълване на отрицатерните числа с предварително определено число.

60. 8) {123} с плаваща, променлива, фиксирана.

60. 9) {125} Възможни са две решения: 1) абсолютна стойност и знаците + и –; 2)допълване на отрицателните числа с предваретилно определено число.

61. 088А. Посочете предимства на правия код при запис на цели със знак.

61. 1) {105} Акуратност в изчисленията. Бързина

61. 2) {106} Първия бит се определя от представянето на знака; диапазон е [–(2n – 1), +(2n – 1)] Не могат да се представят всички числа в прав код

61. 3) {111} Предимства са че събирането на числа с еднакви знаци става по правилата на събиране на двоични числа а знака на сумата съвпада с знаците на числата.
Разликата на числата с различни знаци се свежда до събиране на числата с еднакви знаци

61. 4) {111} първият бит се опр. за представяне на знака; диапазонът е [–(2n – 1), +(2n – 1)]

61. 5) {111} Събирането на числа с еднакви знаци става по правилата на събиране на двоични числа,

61. 6) {112} Предимствата са, че събирането на числа с еднакви знаци става по правилата на събиране на двоични числа, а знакът на съмата съвпада със знаците на числата. Разликата на числа с различни знаци се свежда до събиране на числа с еднакви знаци. Недостатъците са, че има две нули и че при събиране на числа с различни знаци първо трябва до си определи кое от числата е по-голямо **[Че това не е ли числото със знак „+“?]** и от него да се извади другото

61. 7) {113} събирането става по правилата на събиране на двоични числа а знака съвпада при сума. а събирането на отрицателни се свежда до събиране на положителни

61. 8) {114} по-бързо е.

61. 9) {114} Предимство че събирането на числа с еднакви числа, а знаците съвпада със знамения на числата

61. 10) {121} Няма нужда от допълнително кодиране (инвертиране и допълвания), изваждането се свежда до събиране, събирането при еднакъв знак е лесно.

61. 11) {121} Лесно се извършват аритметични действия.

61. 12) {122} Предимствата са че събирането на числа с еднакви знаци става по правилата на събиране на двоични числа а знакът на сумата съвпада със знаците на числата. Разликата на числата с различни знаци се свежда до събиране на числа с еднакви знаци.

61. 13) {124} Сравняването става без да се отчита знака

61. 14) {124} Непосредствено и лесно кодиране
Симетрична стойност на представянията: $2^{n-1}–1$ до $2^{n-1}–1$

61. 15) {125} предимства непосредствено и лесно кодиране, симетрична стойност/диапазон предстaвената –2n–1 до 2n–1.

62. 088В. Посочете недостатъци на правия код при запис на цели със знак.

62. 1) {113} Не могат да се представят всички числа в прав код
62. 2) {113} Недостатъци на правия код са възможност за препълване при опит за запис на отрицателно число
62. 3) {118} Първият бит се отделя за представяне на знака.
62. 4) {118} недостатъци са:
не еднозначното представяне на числата
62. 5) {121} Може да има два записа
62. 6) {125} Не могат да се представят всички числа в прав код
62. 7) {125} цифрата в най-младшия разряд, отговаряща за знака, създава объркване, когато числото запълва 4, 8, 16, 32 и т. н. кутийки от паметта.
62. 8) {126} Първия вид се определя за представяне на знака

63. 089. Как се кодират целите със знак в обратен код при компютрите?

63. 1) {124} В първия разряд се отбелязва знака на числото. Ако е „+“ слагаме 0 а, ако е „–“ слагаме 1. За да завършим представянето, ако числото е отрицателно, трябва да заменим всяка нула с единица и всяка единица с 0.
63. 2) {124} Към числото се добавя единица
63. 3) {125} Целите със знак в обратен код могат да бъдат представени без знак, като изберем половината от тях да бъдат положителни, а останалите използваме за представяне.
63. 4) {125} като 0-та става 1-ца, а 1-цата 0
63. 5) {126} Положителни цели – вземаме положителен знак ← 0(– (– ((– (–) стойност на числото и допълваме. знак ← ((– (– (– (–)
63. 6) {126} Нека pn/2>A≥0
Обратния код се получава като извадим цифрите на А от цифрите на p–1.

64. 089. Как се кодират целите със знак в обратен код при компютрите?

64. 1) {111} като инвертира знаците
64. 2) {111} Нека $2^n/2 > A > 0$. Обратния код се получава като цифрите на |A| се изваждат от цифрата p – 1
64. 3) {121} Кодират се с обратен начин
64. 4) {122} Понеже в p-ична бройна система с n цифри можем да представим pn числа без знак, избираме половината от тях (от 0 до (pn/2 да бъдат положителни а останалите използваме за представяне на отрицателни числа които пък допълваме с (pn–1)

65. 090. Какъв е диапазонът на представимите с n бита цели със знак при обратен код?

65. 1) {110} $2^{n-1} \div -2^{n-1}$
65. 2) {115} $0,2^{n-1}$ или $-(2^{n-1},0)$
65. 3) {115} От $2^n - 1$ до 0
65. 4) {122} $(2^n–1), 2^n–1$ **[Това е страхотен диапазон!]**
65. 5) {122} $[-(2^n–1), +(2^n–1)]$ (
65. 6) {122, 125} [–(2n–1–1); 2n–1–1]
65. 7) {123, 126} От -2^{n-1} до $+2^{n-1}$
65. 8) {123} $[-(2^n–1); (2^n+1)]$
65. 9) {126} от –1 до n и от n до 1

66. 091. Определете операция „допълнение до 1“.

66. 1) {110} „Допълнение до 1“ е операция за избягване на препълване при аритметични действия.
66. 2) {116} **Инвертиране** на числата **в допълнителен код** о

67. 092. За какви цели се използва операция „допълнение до 1“?

67. 1) {104} Да се избегне плаваща запетая (1101262013 Милко Динков Петков)

67. 2) {105} Допълнение до 1 се използва, за да се инвертират битовете на двоично число в обратен код. След това инвертиране се получава числото в обратен код. Операцията се използва за конвертиране на числа със знак от в ОК и обратно.
67. 3) {105} Събиране на числа
67. 4) {105} За получаване на обратен код в допълнителен код.
67. 5) {105} „Допълнение до 1“ използваме при уеднаквяване (закръгляне) на кода.
67. 6) {115} За инвертиране на битовете на думата

68. 094. Определете десетичната стойност на записаните в обратен код двоични числа 110011 и 001101

68. 1) {121} Програмата с по-голям приоритет се изпълнява първа, другите чакат (висящи са)
68. 2) {121} при събиране/изваждане

69. 095. Как се кодират целите със знак в допълнителен код при компютрите?

69. 1) {105} фиксираната запетая означава че мястото на десетичната запетая никога не се мени и е определено
69. 2) {106} Нека $p^n/2 > A \geq 0$ и $A \leq 0$, тогава $A + p^n = ((p^n - 1) - |A|) + 1$, т. е. допълнителният код на целите със знак се получава като към обратния добавим 1.
69. 3) {106} Кодирането на целите със знак се кодират най-често, като 1вия бит на клетката заема 0 за (–) и 1 за (+) знак, съществуват и други видове кодиране.
69. 4) {111} Нека $pn / 2 > A \geq 0$. Нека $A \leq 0$. Тогава $A + pn = ((pn - 1) - |A|) + 1$. Тоест допълнителният код се получава като към обратния прибавим +1.
69. 5) {111} **Същото кодиране като в прав код**, но ако числото е отрицателно след кодирането прибавяме 1.
69. 6) {124} Пред кода се поставя 0 за „+“ и 1 за „–“.
69. 7) {126} – положителни цели – съвпада с изправния код
– Отрицателни цели – като модул на числото
69. 8) {126} – положителни цели – съвпада с изправния код;
– Отрицателни цели –

70. 096. Какъв е диапазонът на представимите с n бита цели със знак при допълнителен код?

70. 1) {110} $–(2n–1); 2n–1–1$
70. 2) {111, 125} $[–2n – 1; 2n – 1 – 1]$
70. 3) {112} $(2n – 1 – 1); 2 n – 1 – 1]$
70. 4) {113} $[–2^{n–1}; 2^{n–1} – 1]$
70. 5) {114} $(1 + 2^{n+15}) – (1 + 2^{n–1})$
70. 6) {124} Диапазонът е $–2^{n–1}–1$ до $2^{n–1}–1$
70. 7) {124} От $–2^{n–1}–1$ до $2^{n–1}–1$

71. 097. Определете операция „допълнение до 2“.

71. 1) {104} Операция допълнение до 2 означава да се добави подходяща стойност, така че крайния резултат да бъде допълнен до 2
71. 2) {105} Към обратния код се добавя 1.
71. 3) {107} Операцията „допълнение до 2“ – е инвертирането на битовете на една дума и 1.
71. 4) {109} Операция е МИ която се извършва от ЦП посредством МЕ
71. 5) {111} На обратния код се прибавя 2
71. 6) {112} Операция което инвентира битове резултат
71. 7) {125} Същата операция като „допълнение до 1 само че с едно повече.

72. 098. За какви цели се използва операция „допълнение до 2“?

72. 1) {111} Допълнение до 2 се използва за образуване на допълнителен код.

72. 2) {111} Операция която инвертира битовете на една дума и добавя 1, **се нарича** допълнение до 2.
72. 3) {121} Операцията допълнение до 2 се използва за преобразуването на обратен в допълнителен код.
72. 4) {123} Операцията допълнение до 2 се използва за за се избегне препълването в пъметта както се случва се опитаме да запишем число с плаваща запетая
72. 5) {123} операцията инвертира битовете на една дума и прибавя 1
72. 6) {125} операция която инвертира битовете на една дума и прибавя 1

73. 101.Какво представляват модифицираните обратен и допълнителен код?

73. 1) {111} За да се установи дали има **препълване след събиране на числа с еднакъв знак**.
73. 2) {121} Модифицирания и обратния код представлява начин за проверка и модификация на процеси и операции.
73. 3) {121} Обратен Код обърнатата стойност на кода.
73. 4) {123} Дублирането на знака се реализира преди сумиране, а сигнализация за препалване е различие в знаковите битове на сумата
73. 5) {124} Обратен са превръща двоичните числа в обратния им ред, в допълнителен се представат числата в десетична стойност.
73. 6) {125} Дублирането на знака се извършва преди сумиране а сигнализация за препълване е различие в знаковите битове на сумата.
73. 7) {125} Обратен код → инвертираен на битовете на ПК
Допълнителен код → прибавяне на 1 към ОК.

74. 102. За какво се използват модифицираните обратен и допълнителен код?

74. 1) {121} Обратен код и допълнителен код са създадени за да може двоичният код да бъде пресмятан лесно т. е. операциите за събиране изваждане умножение и деление да могат да се извършват и при двоичен коз.
74. 2) {122} За да се **останови** дали има препълване след събиране на числа с еднакви знаци
74. 3) {123} Модифицираните обратен и допълнителен код се използват записването на числа от десетична бр. с-ма в двоична
74. 4) {126} Дублирането на знака се извършва преди сумирането, а сигнализация за препълване е различие в знаковите битове на сумата.
74. 5) {126} За да се установи дали има препълване след събиране на числа с еднакви знаци

75. 105. По какъв начин могат да се представят дробните числа в компютрите?

75. 1) {104} Дробните числа в компютрите могат да се представят чрез плаваща запетая.
75. 2) {104} Дробните числа в компютрите могат да бъдат представени като се преобразуват в число с десетичен остатък (пример: 4/10 = 0,4)
75. 3) {105} Със знака „/“ (¾ = ¾).
75. 4) {105} Дробните числа в компютрите могат да се представят, чрез знаците на клавиатурата или по-точно чрез „,“ – запетая и чрез функцията „Math.
75. 5) {110} Представят се двоично, с плаваща запетая, с допълнителен код.
75. 6) {120} Дробните числа могат да **се представят в десетичен вид** като има две степени на закръгляване:
1-вия е до 7 знака след запетая „,“ .
II-рия е до 14знака след запетаята „,“
75. 7) {122} С плаваща запетая.
75. 8) {123} Използват се **три** варианта за представяне на дробни. **1** – с фиксирана запетая **2**– с **естествена фиксирана** запетая
75. 9) {124} Дробните числа, могат да бъдат представени като изместен код с плаваща запетая.
75. 10) {124} Чрез съвкупност от 0 и 1.
75. 11) {124} Числото се разделя на две, едната част е стойност преди запетаята, в другата след запетаята

76. 106. Как се кодират дробните числа при представяне с фиксирана запетая?

76. 1) {101} За да увеличим диапазона на представимите числа трябва да пожертваме броя на верните цифри в записа
76. 2) {105} Числата преди запетаята **се кодират на цели числа**, а след запетаята – остатък
76. 3) {107, 123} мястото на запетаята никога не се мени т. е. е фиксирано
76. 4) {110} С помоща на прав код.
76. 5) {119, 123} Дробните числа се кодират с двоичните цифри.
76. 6) {123} Като се използва позиционна бройна система
76. 7) {125} Чрез мантиса и порядък.

77. 108. Къде най-често се фиксира двоичната запетая при компютрите?

77. 1) {109} В изпълнителните операции защото там е необходима точност а не бързодействие.
77. 2) {114} След втората цифра
77. 3) {122} Двоичната запетая се фиксира в мантисата, която представлява предварително определена по брой поредица от битове, всичко преди нея е цялата част от числото, а след нея е дробната част.
77. 4) {124} Запетаята се фиксира най-често в лявата част на двоичните записи. В едно число близко до 0 запетаята се фиксира непосредствено след първият ненулев елемент от ляво на дясно. 0,1 в три бита = $1,0001_{(2)}$.

78. 109. Как се кодират дробните числа при представяне с естествена запетая?

78. 1) {102} Дробните числа се кодират, като се раздели променливата на две части – цяло число и десетична част. Например, ако променливата е 16 бита 8 бита остават за цялото число и 8 за десетичната част.
78. 2) {103} Мястото на запетаята никога не се мени т. е. фиксирано. Част от битовете в думата са цялата част на числото а останалите дробни
78. 3) {116} Мястото на двоичната запетая никога не се мени – то се опр. предварително. Част от битовете на думата са цифрите в цялата част на числото, а останалите в дробната.

79. 111A. Къде най-често се използва представяне с естествена запетая?

79. 1) {105} В правият код.
79. 2) {122} Най-често се използва при дробните числа.
79. 3) {123} При представяне на данни в ОП по-конкретно при представяне на дробни числа.
79. 4) {124} Където искаме да закръглим до –1.

80. 111B. Защо при калкулаторите най-често се използва представяне с естествена запетая?

80. 1) {104} При калкулаторите най-често се използва представяне с естествена запетая за по-голяма точност при изчисление.
80. 2) {104} Да се избегнат грешки
80. 3) {105} Защото позицията на естествената запетая може точно да се определи (зададе) за нуждите от точност на изчисленията
80. 4) {105} Увеличаване на диапазона.
80. 5) {105} Защото при представянето естествена запетая едно от числата посочва къде да се постави запетаята, което дава възможност за много точно представяне на числа с много цифри в десетичната част
80. 6) {105} При калкулаторите най-често се използва представяне с естествена запетая защото изчисленията се извършват по-бързо.
80. 7) {106} Защото при това представяне имаме фиксирана грешка (абсолютна), докато при представяне с плаваща запетая имаме относителна грешка и делението е т. нар. „опасна“ операция.
80. 8) {111} Защото калкулаторите са с фиксирана точност на пресмятане.
80. 9) {111} Защото при калкулаторите диапазонът на числата е по-голем

80. 10) {112} Представянето на числата от десетичната бройна система е знаково, а при изчисляването им в двоична бройна система и преубразуването им в десетична се налага **закръгляване с точност**.
80. 11) {112} За да пресмята вярно
80. 12) {121} Заради предварително фиксираната точност
80. 13) {121} Защото там е необходимо да се използват числа с много голям интервал.
80. 14) {121} Заради големия диапазон на числата.
80. 15) {122} Защото мястото на запетаята е предварително зададена от потребителя.
80. 16) {122} Заради големият диапазон на числата
80. 17) {124} За по-лесно разбиране и представяне.
80. 18) {124} При този вид представяне има голяма точности не съществуват проблеми при операция деление.
80. 19) {125} За да се увеличи точността

81. 112. Как се кодират дробните числа при представяне с плаваща запетая?

81. 1) {105} Дробните числа се кодират посредством поставянето на плаваща запетая след третия знак.
81. 2) а да увеличим диапазона на представяните числа трябва да пожертваме броя на верните цифри в записа. Нека x > 2 и p ? 2. Тогава x = m ? pe за някой m и e цяло число. Такъв запис на числата се нарича експоненциален.
81. 3) {115} Всяко число се представя с експоненциален запис, мантисата се отбелязва с „m“, порядъка с „e“, а с „p“ се отбелязва основата на позиционна бройна система: $x = m.p^e$

82. 113. Какво определя броят на цифрите за запис на мантисата при плаваща запетая?

82. 1) {106} Определя цифрите след запетаята
82. 2) {106} При плаващата запетая имаме к-цифри една част к-цифри, която определя броя на целите числа **[Че те не са ли безброй много?]**, а друга част определя броя на дробните числа **[Нима и те не са безброй много и то повече от целите?]**.
82. 3) {111} Диапазонът на числата, които се представят.
82. 4) {116} дължината на мантисата
82. 5) {119} До кой знак след запетаята да се запише числото.
82. 6) {121} Определя знаменателя на числата. дробна черта, числата след нея.
82. 7) {122} Диапазона на представяните числа
82. 8) {124} Броят на битовете при записване на числото.
82. 9) {124} Броят на цифрите след запетаята.
82. 10) {125} Броят на реалните цифри
82. 11) {125} порядакът.

83. 114. Какво определя броят на цифрите за запис на порядъка при плаваща запетая?

83. 1) {105} Мантисата и експонента.
83. 2) {105} Броят на цифрите на порядъка определя големината на числото.
83. 3) {105} Порядъка при плаваща запетая определя броят на цифрите при което след пресмятане няма загуба на данни и доп. грешка е минимална
83. 4) {105} Точността на числото с ПЗ (Мантиса)
83. 5) {109} Големината на зададената памет за записа
83. 6) {111} експонента
83. 7) {111} Броят на цифрите за запис определе **точността на порядъка – диапазона**.
83. 8) {112} Точността, броят верни цифри.
83. 9) {113} Разрядността на кода
83. 10) {114} **порядъкът** определя **точноста на броя на цифрите**
83. 11) {114} Мантиса
83. 12) {116} Броят на цифрите на мантиката определят точността (брой верни отговори).
83. 13) {121} Диапазона на дробната част

83. 14) {122} Броят на цифрите за запис определя точността на порядъка – диапазона.
83. 15) {122} Определя точността на записаното число след запетаята
83. 16) {123} броят ня цифрите за запис на порядъка при плаваща запетая определят точността на числото и до кой знак ще бъде закръглено.
83. 17) {124} Точността – броят верни цифри
83. 18) {124} Броят на цифрите определя числото. а мантисата степента.
83. 19) {125} Броят на цифрите за запис на порядъка при плаваща запетая определя точността на порядъка – диапазона. **[Така формулиран отговорът е твърде неясен!]** .
83. 20) {125} Броят на цифрите за запис орледеля точността ва записа диапазона

84. 115A. Защо е необходимо нормализираното представяне в плаваща запетая?

84. 1) {105} При събиране на числа с плаваща запетая с различен порядък
84. 2) {106} Защото работата с него е по-лесна и за по-лесно и точно пресмятане.
84. 3) {115} За да се избегне препълване.
84. 4) {115} Защото се повишава точността на представеното по този начин число.
84. 5) {115} За да се избегне препълване
84. 6) {115} За да се пести памет

85. 115B. Какво предвижда нормализираното представяне в плаваща запетая?

85. 1) {105} еднозначност при $x \neq 0$
85. 2) {110} По-точни изчисления от целите числа
85. 3) {111, 111} **Необходимо е**, защото представянето с плаваща запетая не е еднозначно. Нормализираното представяне предвижда **първата цифра** на мантисата **да е винаги 1**.
85. 4) {111} Нормализираното представяне предвижда първата цифра на мантисата да е винаги 1
85. 5) {121} Необходимо е защото представянето с плаваща запетая не е еднозначно. Нормализи–ранеото представяне предвижда първата цифра на мантисата да е винаги 1
85. 6) {121} При нормализираното представяне плаващата запетая трябва да е точно един бит от цялата поредица от информация.
85. 7) {122} Еднозначност при x 0: 1/p ≤ m (първата цифра на мантисата да бъде = 0) (
85. 8) {122} Еднозначност при x=0 1/p s|m| < 1 – нормализирано представяне първата цифра на мантисата да бъде = 0
85. 9) {122} Нормализирането предвижда първата цифра в мантисата да е 1
85. 10) {123} Лявата част е цялото число, дясната – дробната част.
85. 11) {123} За да бъде мантисата нормализирана е необходимо първата и цифра да бъде ≠ 0.
85. 12) {125} Необходимо е, защото представянето с плаваща запетая не е еднозначно.
85. 13) {125} Предвижда първата цифра на мантисата да е 1 всеки път.
85. 14) {125} Нормализираното представяне представлява предвиждане които прави, така че първата цифра на мантисата винаги да бъде 1.

86. 116. Какво е особеното на нормализирано представяне в плаваща запетая при основа 2?

86. 1) {105} Числата се представят чрез техните цифри

87. 117A. Как се използва особеността на нормализираното представяне в плаваща запетая при основа 2?

87. 1) {106} При нормализирано представяне в плаваща запетая при основа две първата цифра на нормализираната мантиса винаги е 1. При p = 16 нормализираната мантиса може да започва с до три 0
87. 2) {112} При p = 2 първата цифра на нормализираната мантиса винаги е 1
87. 3) {113} Първата цифра на мантисата е винаги 1.
87. 4) {114} Мантисата е 1.
87. 5) {122} (~~Продвижда еднозначност~~) При различни порядъци трябва да се денормализира една мантиса за да се извършат порядъците и да се извърши събиране.

87. 6) {122} Първата цифра на мантисата е винаги 1.
87. 7) {125} Първото число на мантисата е винаги 1

88. 118. Защо при събиране с плаваща запетая не се говори за „препълване“?

88. 1) {102, 103, 113} При равни порядъци можем да съберем мантисите, а при различни трябва денормализиране една мантиса.
88. 2) {103} При събиране с плаваща запетая не говори за „препълване“ защото при равни порядъци можем да съберем мантисите, а при различни порядъци трябва да денормализираме едната мантиса.
88. 3) {103} Защото при препълване просто увеличаваме мантисата, а когато мантисата не достига за представяне на числото се получава безкрайност.
88. 4) {111} при равни порядъци можем да съберем мантисите,а при различни трябва да денорма–лизираме едната мантиса.
88. 5) {111} Защото това **[Кое е това „това“?]** се избягва с нормализирането на мантисата.
88. 6) {114} Защото при събирането с плаваща запетая препълването е заместено с „Машинна 0“.
88. 7) {114} Вторият начин е чрез маскиране на заявките.
88. 8) {122} Защото при препълване увеличаваме мантисата. и когато тя не стига за представяне на числото се получава безкрайност.
88. 9) {124} Защото излишните нули от кода се премахват.
88. 10) {125} Масивна нула
88. 11) {125} При равни порядъци можем да съберем мантисите а при различни трябва да денормализираме едната мантиса
88. 12) {125} Не се говори за препълване, защото данните са от тип fbat, който е по-голям а резултатът не е цяло число.
88. 13) {126} Защото при препълването просто увеличаване мантисата, а когато мантисата не достига за представяне на числото се получава безкрайност

89. 119. Кой термин замества термина „препълване“ при плаваща запетая и защо?

89. 1) {103, 126} Денормализиране на едната мантиса, за да я съберем с другата, ако са с различен порядък
89. 2) {103} Безкрайност. Защото при препълване се увеличава мантисата, а когато това се случи числото става безкрайност
89. 3) {104} „нули“ – защото закиченото число може да изгуби стойността си и да стане нула
89. 4) {111, 111, 113, 114, 114, 119, 121, 122, 122} Машинна нула.
89. 5) {111} Денормализиране на едната матрица, за да я съберем
89. 6) {111} Това е терминът „Машинна нула.
89. 7) {113} (Carry, CF) = препълване без знак
89. 8) {115} Машинна 0, защото числото е граничещо че кагонещо към 0
89. 9) {115} Безкрайност
89. 10) {122} Терминът, заместващ „препълване“ наричаме **рекурсия**, защото при препълването числата след запетаята достигат максималния брой.
89. 11) {124} Закръгляне, защото числата с плаваща запетая не могат да се представят с точност в двуична бройна система.
89. 12) {125} Диапазонът на представимите цели със знак при допълнителен код е -2^{n-1} до $2^{n-1}-1$
89. 13) {125} Безкрайност. Защото числото реално не може да се препълни, а става безкрайно голямо

90. 120. Какво представлява понятието „характеристика“ при някои системи с плаваща запетая?

90. 1) {105} Понятието характеристика отразява свойствата на дадена система
90. 2) {105} През определен интервал се слага запетая, който се величава.

91. 121. Защо в някои системи с плаваща запетая вместо порядък се използва характеристика?

91. 1) {113} за да бъдат избегнати грешки.
91. 2) {114} действието делене е рисково

92. 122. Какво е покритието на реалната ос при представяне на дробните числа с фиксирана запетая?

92. 1) {115} Неравномерно

93. 123. Какво е покритието на реалната ос при представяне на дробните числа с плаваща запетая?

93. 1) {114} Закръгля да па близката част цяло число

94. 124. Кой вид грешка е постоянна при представяне с фиксирана запетая? Защо?

94. 1) {121} При фиксирана запетая постоянна грешка се получава при операция деление, защото се губи дробната част.
94. 2) {122} **Препълване**
94. 3) {124} Абсолютната грешка **[Защо?]**
94. 4) {124} С фиксирана запетая се появява постоянна греша когато се извършва операция събиране защото числата до запетаята има голем може да се обърка
94. 5) {124} Не точното представяне на информация
94. 6) {124} Относителна поради неравномерното разпределение на числата върху числовата ос.
94. 7) {125} Има постоянна грешка при делене, губи се част от дробната част.

95. 125. Кой вид грешка е постоянна при представяне с плаваща запетая? Защо?

95. 1) {108} Фиксирана абсолютна грешка. При относителна – не абсолютна.
95. 2) {111} Губенето на мантисата е постоянна грешка при представяне с плаваща запетая.
95. 3) {111} Абсолютната – поради равномерното разпределение на числата върху числовата ос.
95. 4) {121} Разполагаме с краен брой битове при което имаме загуба на точност.
95. 5) {122} Определянето на броя цифри след запетаята и пресмятането им правилно.
95. 6) {123} Абсолютната грешка.
95. 7) {123} Унищожаването на дробната част (след запетаята) или на част от нея което води до неточен резултат.
95. 8) {124} Грешка за точност.

96. 126. Коя аритметична операция е „опасна“ при представяне на числа с фиксирана запетая и защо?

96. 1) {104} При изваждане, защото има „опасност“ от объркване или изгубване на запетаята.
96. 2) {104} Умножението
96. 3) {105} умножение, защото при тази операция имаме значителна загуба от порядъци.
96. 4) {105} Събиране
96. 5) {105} При умножение има опасност от изместване на фиксираната запетая.
96. 6) {114} Делене (загуба на дробната част) събиране на близки по абсолютна стойност числа с различни знаци **[Така написан отговорът е нееднозначен и абсолютно грешен!]**

97. 127. Коя аритметична операция е „опасна“ при представяне на числа с плаваща запетая и защо?

97. 1) {105} Аритметична операция „машина нула“ е опасна при представяне на числа с плаваща запетая.
97. 2) {106} „Опасна“ аритметична операция е деленето, защо при обратен код изчислението не е правилно. Загуба на дробната част.
97. 3) {112} Деление

97. 4) {114} При представяне на числа с плаваща запетая е опасна операция събиране, защото могат да се получат неверни резултати

98. 129. Какви са неудобствата на двоичната бройна система на компютрите за работата на хората?

98. 1) {104} Двоичната бройна система е неудобна за хората понеже колкото по големи стават числата толкова по голямо е двуичното му изражение. Освен размера главен проблем е и читаемостта единственото лесно за възприемане е четност или нечетност. Също така ДБС предполага и бързи пресмятания на сборовете на квадратите на 2, което за някой е тежка задача.
98. 2) {105} Неудобствата са свързани най-вече с това, че сме свикнали да използваме десетична бройна система. Неяснота при изписването. Необходими са изчисления
98. 3) {105} Неудобствата се изразяват в това, че всички числа представени в двоичен вид са много дълги и това затруднява работата с тях
98. 4) {105} Неразбираема за хората
98. 5) {105} Неясност при изписването; мат повече място
98. 6) {107} По-бавно изпълнение при работа с двоичната бройна система.
98. 7) {107, 116} Много дълъг запис на често използвани числа
98. 8) {110} Трудно възприемане на числата само с 0 и 1, Прекалено дълго за изписване и за запомняне дадено двуично число.
98. 9) {114} липсва вход и изход
98. 10) {114} Двоичната бройна система е непонятна за съвременния човек. Свикнали сме с десетичната бройна система.
– дълга и неудобна за запис и разчитане.
98. 11) {115} Преобразуването то други бройни системи
98. 12) {115} Трудно разбираема е за хората.
98. 13) {116} трудно пресмятане
98. 14) {119} Необходимо е пресмятане и отнемат повече място
98. 15) {119} По-труден вход и изход на данни, Ограничен брой числа (групирани в байтове).

99. 130. Как се елиминират неудобствата на двоичната бройна система при компютрите?

99. 1) {106} Като се превръщат от двоична в десетична, шестнадесетична и т. н. Чрез компилиране.
99. 2) {115} В устройството на вход и изход се борави с реални числа докато компютърът обработва числа от двоична
99. 3) {119} Данните се записват в поредици от осем бита, това изгражда байт, който е по-четлив за програмистите.

100. 131. Какво представляват двоично-кодираните десетични числа?

100. 1) {105} Двоично кодираните десетични (Binary Coded Decimals) – BCD, използват се за улеснение на работата при използване на **десетична и двоична** бройна система.
100. 2) {105} ДКД са два вида пакетиране и непакетирани. Едно число се записва в един полубайт Цифрите са в диапазона 30÷39 шестнадесетично или F0÷F3. При пакетиран код в двата по–лубайта има по една Цифна
100. 3) {105} Двоично кодираните десетични числа се изписват с по четири числа (от 0 до 1) всяко едно число от десетичните числа.
100. 4) {107} Това е представянето на десетичните числа в двоична форма, с която работят компютрите. Във вид на единици и нули
100. 5) {114} Числа представени като съвкупност от единици и нули
100. 6) {115} Запис на десетично число в двоична ПБС
100. 7) {120} Десетични числа се използват за входно/изходни операция в по модерните компютри, те се кодират в двоичен коз за да може ЦП да извърши по-бързо пресмятанията.

100. 8) {123} Десетичните числа се представят с четири цифри – двоично кодирани десетични числа

101. 132. Какви варианти за представяне на ДКД числа познавате и как се наричат числата при всеки от тях?

101. 1) {112} Има няколко системи за реализиране на работата с ДКД числа: 1) IBM–360: задаване на броя на цифрите и знака в пълен набор от МИ: 2) 6502 два режима на работа на АЛУ – двоичен и десетичен за ADD и SUB 3) I80x86 и др.

101. 2) {115} 8 битова клетка (байт), 2 възможности, неопаковани ДКД

102. 133. При наличие на двоичен суматор, необходимо ли е да се разработва отделен суматор за работа с ДКД числа? Защо?

102. 1) {102} Да, необходимо е! Защото двоичния суматор неможе да работи с ДКД числа.

102. 2) {104} Да, За обработка на постъпваща информация

102. 3) {105} Да, за да може числата да се преобразуват, тъй като двоичният суматор не работи с ДКД числа.

102. 4) {118} При наличие на двоичен суматор е необходимо да се разработи и един за ДКД за да може да конвентира числата в двуичният им вид при употреба на ДКД.

103. 134. Как се представят знаковете, с които пишем думите, при компютрите?

103. 1) {105} Това последователност от нули и единици

103. 2) {106} Представят се с код, който е универсален за всеки символ от думите и знаците, които използваме.

103. 3) {114} Низ от букви, цифри и специални знаци

104. 135. Какво представлява понятието „знаков код“?

104. 1) {105} „Знаков код“ е средството чрез което могат да се представян знаци, букви, като за всеки знак отговаря определена цифра

104. 2) {112} Знаковият код е съвкупност от символи, организирани по начин, лесен за обработка от ПК.

105. 136. Какви принципи трябва да се спазват при създаване на знакови кодове?

105. 1) {102} Точна номерация, да няма повторения, знаците, които имат големи и малки вариации да се записват по отделно

105. 2) {102} Знаковите кодове трябва да са представени от цели положителни числа и да нарастват с предвижване Напред в знаковото множество. Пр. ASCII (a)= 52, (A) = 82

105. 3) {104} Логическите принципи на езика

105. 4) {105} Знакови кодове имат точни и ясни принципи които трябва да се спазват за да работи опр. програма или алгоритъм

105. 5) {114} Знакови кодове имат точни и ясни принципи които трябва да се спазват работа

105. 6) {123} Когато се създават знакови кодове трябва да се има предвид, че кодовете трябва да могат да описват всичките МИ които се изпълняват

106. 137. Защо цифрите (0–9) трябва да се кодират с последователни цели числа?

106. 1) {105} За да може да се извършват аритметични действия с числати.

106. 2) {106} Защото трябва да се представят под формата на т. нар.машинен език или по точно казано под формата на **двуична** бройна система

106. 3) {109} Защото компютрите могат да разберат само 2 състояния (има ток и няма ток). Следователно цифрите от 0 до 9 са представени чрез двоичната бройна с-ма (0 и 1-ци)

106. 4) {110} За да могат да бъдат прочетени като двоично кодирани десетични числа.

106. 5) {115} Предполагам за да може по-лесно да се правят изчисления със съставени от тях число,

106. 6) {115} За да може лесно да се сравняват цифрите на числата

106. 7) {115} Цифрите (0–9) трябва да се кодират, защото в противен случай се губи възможността за директно сравнение и намиране на разликите между тях.
106. 8) {116} За да може лесно да минаваме от знак към цифра с цел извършване на аритметична операция
106. 9) {118} Защото се губи директното сравнение по между си.

107. 138. Защо числата, кодиращи буквите, трябва да се увеличават, съгласно реда на буквите в азбуката?

107. 1) {105} Защото има по-голяма стойност в бройната система
107. 2) {110} За да са по-разбираеми и достъпни за хората както са си подредени по принцип
107. 3) {114} Числата, кодиращи буквите, трябва да се увеличават съгласно реда на буквите в азбуката, за да се избегне грешка

108. 142. Как става обработката на аналогови данни при цифровите компютри?

108. 1) {105} Обработката става чрез конвертиране на аналог. в цифра данни.
108. 2) {105} Аналоговите данни се конвертират в цифрови, за да бъдат използвани от цифровите компютри
108. 3) {105} Обработката на аналоговите данни се осъществява чрез физични явления като магнетизъм
108. 4) цифрови данни на входа и изхода за по-голяма точност, аналогови елементи за по-голяма точност на изчисление.
108. 5) {115} Като аналоговите данни се представят чрез цифри
108. 6) {115} Като се преобразуват в специфична поредица от 0 и 1
108. 7) {116} Аналоговите данни се „дигитализират“ **[Българино знай своя род и език!]** преди обработка в цифровите компютри, т. е. работи се с техния цифров еквивалент.
108. 8) {116} Като се преобразуват специфична поредица от нули и единици
108. 9) {119} **Чрез представянето** им в дискретен вид, т. е. като поредица от цифри
108. 10) {119} В системите за цифрова обработка, сигналът винаги е представен до определено количество битове.
108. 11) {119} Чрез представянето им в дискретен вид.

109. 143. Какви варианти за представяне на изображения се използват при компютрите?

109. 1) {105} Векторни

110. 144. На какво се основава растерното представяне на изображения?

110. 1) {106} Растерното изображение се основава на точкуването. Създават се изображения на база потъмняване на определени пиксели от екрана. Така се изгражда изображение в човешкия мозък.
110. 2) {106} Растерното представяне на изображението се основава на човешкото възприемане (зрение)
110. 3) {115} Растерните изображения се състоят от много на броймалки точки (пиксели) с определен цвят, които окото възприема като цял образ.

111. 145. Как се представят точките при растерно изображение?

111. 1) {107} Чрез кодиране на цвета на точката
111. 2) {110} Растерното изображение използва недостатъците на окото.
111. 3) {115} Като цифра (0 и 1)
111. 4) {116} чрез копиране на цвета ва точката в 1 bit за черно бяла, 256 bit за оттенък на черно, 1 byte за син зелен и червен.

112. 147. На какво се основава векторното представяне на изображения?

112. 1) {102} Векторното представяне на изображения представлява изображение представено с вектори. Това позволява изображението да запазва качеството си при по-големи резолю-

ции. „Векторните изображен сеизползват в предпечата. Растерните изображения са другия вид

112. 2) {105} На пикселиране Конкретни пиксели на екрана биват потъмнявани. В програмата са зададени точно кои пиксели (адреси) трябва да бъдат потъмнени

112. 3) {105} Векторното представяне на изображения се основава на смесването на три основни цвята.

112. 4) {107} При цифровите компютри изображения се представят растерно и векторно. Векторното изображение се представя матрични процесори, който оперират на множество данни

112. 5) {110} На недостатъците на човешкото око.

112. 6) {110} **На художническата четка**. Като се рисуват криви с различна дължина.

112. 7) {114} На растерни изображения

112. 8) {115} Свързването на две точки с криви, като предварително е взет размера (на четката

113. 148. На какво се основава представянето на звук в компютърните системи?

113. 1) {104} Представянето на звук в компютърните системи се основава на няколко основни музикални тона, които съставят звук

113. 2) {105} Човешкото ухо различава честоти от 16 Hz до 20 KHz. За да се представи звук по цифров път е достатъчно да се отчита амплитудата на сигнала, измелена през двойно по-малък интервал, т. е. с двойно по-голяма честота. Честотата, през която се измерва тази амплитуда е ~ 44 kHz.

113. 3) {105} Звукът представлява вълна, емулира се от компютъра посредством тонколони на които се подава сигнал с различна честота (max 48 kHz). Звуковият файл е последователност от дигитализирани честоти във времето

114. 149. Как по цифров път се представя звук?

114. 1) {105} Звукът се представя по цифров път с помощта на цифрови устройства. Използва се недостатъка на човешкото ухо

114. 2) {105} Човешкото ухо улавя звукови вълни в диапазона 20 Hz – 20 KHz. За постигане на вярно възпроизвеждане при компютрите се използва т. нар. квантоване по време на аналоговия звуков сигнал с двойно по-голяма честота до 41 kHz

114. 3) {110} Аналогов сигнал

115. 150. Какви схеми за цифрово представяне на звук познавате и по какво се различават те?

115. 1) {110} Различават се по количеството данни, които съхранява всеки „семпъл“.

115. 2) {110} За цифрово представяне на звук се използва дефекта на човешкото ухо да разпознава звук от 20 Hz до 20 kHz. **Амплитудата се** измерва и **променя** за двойно по-кратки интервали и така се предава звука.

116. 151. Какво представляват компютърните програми?

116. 1) {115} Компютърните програми представляват алгоритми_ Най-общо – подаваме входяща информация задаваме някакви параметри на специфичен програмен език, като искаме компютърът да обработи дадена информация и да ни върне резултат.

116. 2) {115} Алгоритъм които е в кодиран вид и е съхранен в ОП се нарича компютърна програма.

117. 152. От какво се изгражда една машинна програма?

117. 1) {102} От код, който представлява **алтуритъм**.

117. 2) {102, 121, 122, 123, 124} От код, който представлява алторитъм.

117. 3) {105} Една машинна програма се изгражда от програмен код

117. 4) {106} Изгражда се от код, който представлява алторитъм.

117. 5) {110} Програмен код **и транслатор превръщащ кода в машинен**, посредством 0 и 1.

117. 6) {110} От машинен код създаващ машинни инструкций

117. 7) {111} Изгражда се от **машинен код и инструкции**

117. 8) {114} При него се работи с тунелно пробивна машина. То се придвижва от система хидравлични крикове.
117. 9) {115} От описанието на елементарни действия, нар. машинни инструкции. Когато алгоритъма трябва да се изпълни.
117. 10) {119} Една машинна програма се изгражда като представя, кодира, **изпълнява** алгоритъм.
117. 11) {119} От кодирания вид на алгоритъма.
117. 12) {124} От алгоритъм които ЦП трябва да извърши.
117. 13) {124} Алгоритъм, който трябва да бъде изпълнен, записан в кодиран вид в паметта.
117. 14) {125} От машинен език.

118. 153. Какво представлява понятието „машинна инструкция“?

118. 1) {113} Процесът да протича при строго определен ред.
118. 2) {114} Машинна инструкция

Когато пишем на даден език независимо от езика за програмиране например на Java въвеждаме информацията за нас на машинен език т. е. използвайки елементите на дадения език, а компилаторът превежда на машинен език издавайки машинни инструкции

118. 3) {114} Съвокупносто на машинните инструкции, които разпознава централния процесор
118. 4) {122} Една операция, която се подава от централния процесор.
118. 5) {123} **Код** от ниско ниво, **който се пише дириктно , върху процесора**.
118. 6) {124} Елементарно действие **[На кого?]**
118. 7) {124} Следващата стъпка (код) който трябва да се изпълни
118. 8) {125} Като бъде зададен даден набор от действия за изпълнение, и начина по, който трябва да бъдат изпълнени от машината.

119. 154. Какво представлява понятието „машинен език“?

119. 1) {105} Машинен език е всеки език, който компютъра разпознава и работи с него.
119. 2) {105} Езикът по който се комуникира между входно-изходните устройства и ЦП.
119. 3) {106} **Софтуерна програма** служещ за задаване на команди към компютъра за изпълнение
119. 4) {106} Машинният език представлява последователност от МИ представени в чифри (0 и 1), които ЦП може да разчита.
119. 5) {111} Машинният език се състои от 1 и 0. Той **контактува с ЦП без посредник**.
119. 6) {113} Машинния език представлява език от ниско ниво предназначен за обработка на данни на машинно ниво.
119. 7) {113} Машинния език е език който контактува пряко с ЦП
119. 8) {121} Машинният език е набор от машинни инструкции, които се изпълняват последователно за да извършат дадени операции.
119. 9) {121} Машинен език е език разбираем от компютърната система, непонятен или трудно разбираем за човека
119. 10) {122} Програма, която се изпълнява от компютъра се представя като 0 (нули) и 1 (единица), те се запазват в паметта, компютъра ги чете, те се представят като битове, точно тези битове се наричат машинен език
119. 11) {122} Действителна **програма, която ком. изпълнява** в последователност от единици и нули записани в паметта на комп.
119. 12) {123} Език за програмиране
119. 13) {123} Машинен език – е език с който хората могат да напишат какво точно машината да направи
119. 14) {124} Множество от нули и единици
119. 15) {125} Последователност от битове, възприемащи се от ЦП като алгоритъм, с всички машинни инструкции

120. 155. Какви са компонентите на една МИ?

120. 1) {102} Всяка МИ трябва да посочи две неща 1) Какво трябва да се свърши +, – и др. 2) С какво ще се извърши действието

120. 2) {105} Компонентите на една машинна инструкция в съвременните езици са:
еднокомпонентни
многокомпонентни
Еднокомпонентните биват:
– регистрови
– абсолютим
– непосредствен операнд
– косвени
– автоувеличение
– автонамаление
– косвено автоувеличение
– косвено автонамаление
Многокомпонентните биват:
– страничен
– базови
– индексни
– относителни
120. 3) {105} МИ се състои от код, операнди.
120. 4) {106} – входни данни
– изходни данни
120. 5) {110} – Дънна платка
– Процесор
– ОП.
120. 6) {115} – изпълнение
– регистриране
– пренос на данни

121. 156. Какво посочва кодът на операция на една МИ?

121. 1) {107} Кодът на една операция в МИ посочва нейната форма.
121. 2) {112} Адреси и регистри
121. 3) {115} Показва адресите на клетките които подлежат на обработка.
121. 4) {121} Знакът на съответ операция
121. 5) {124} командите които трябва да извърши ЦП.
121. 6) {124} Над кои данни да бъде извършена съответната операция.

122. 157. Какво посочва адресната част на една МИ?

122. 1) {107} Посочва брой на нейните адресни полета
122. 2) {107, 111} адресността на дадена машинна инструкция посочва брой на нейните адресни полета
122. 3) {111} Посочва **регистъра, над който** ще се извършва операция.
122. 4) {113} Адресността на даден машина инструкция посочва брой
122. 5) {113} Как операция да бъде извършена с подадените данни.
122. 6) {114} Над кои данни ще бъде извършена сътветната на адресация
122. 7) {124} Адресната част на един МИ посочва къде са данните в ЦП.
122. 8) {126} Адресната част на всяка **от тях** се поделя на отделни полета, наречени адресни полета.
122. 9) {126} Адрсността на дадена машина посочва броя на неините адресни полета.

123. 158. От какво се състои адресната част на една МИ?

123. 1) {102} Тя се състои от 2 части – обработващи за извършване н определени пресмятания: събирани и изваждане
адресни полета

123. 2) {113} От 3 компонента – адрес на операнд, адрес за запис на резултат и адрес, сочещ следващата МИ.
123. 3) {119} приемник, източник
123. 4) {119} Над кои данни ще бъде извършена съответната операция
123. 5) {121} Адресната част на една Машинна инструкция се състои от единици наречени битове и 8 → бита → 1 байт. Адресната част е масив от тези елементи.
123. 6) {122} Адресността на дадена машинна инструкция посочва броя на нейните адресни полета.
123. 7) {123} От битове, които определят диапазон, адреси … чрез стойностите си и местоположението си.
123. 8) {124} адресната част на МИ е мястото в паметта, където се съхранява стойността на инструкцията.
123. 9) {125} От адреса на клетките в ЦП към ОП.
123. 10) {125} Адресността на дадена машинна инструкция посочва броя на нейните адресни полета

124. 159A. Защо в едно АП е най-естествено да бъде записан пълен адрес от ОП?

124. 1) {105} Защото адресите на клетките в ОП са идентични (аналогични), затова за да е ясно за коя клетка се отнася е нужно записването но пълния адрес.
124. 2) {106} В едно АП се записва пълен адрес от ОП, за да се избегне възникването на грешка.
124. 3) {106} За да се намира точната информация и да се стига най-бързо до нея
124. 4) {110} За по-бърз достъп до ОП.
124. 5) {111} Защото процесорът обработва междинни данни
124. 6) {113} Поради възможността да запишем много адреси да го търсим и тай е записан с не пйлун яспук няма да можем да го намерим а с пълня
124. 7) {122} Адресните полета (АП) служат за съхраняване на адреси. Най-естествено е записване на пълен адрес от ОП за да може всяко адресно поле да бъде пълно и функционално.
124. 8) {124} Данни

125. 159B. Какво е най-естествено да бъде записано в едно АП?

125. 1) {113} информация с числа
125. 2) {114} операнда

126. 160. Какви видове МИ има? Защо?

126. 1) {105} Видове МИ – управляващи, обработващи, защото това са двете основни функции на машината инструкция – какво ще се прави КОП – къде ще се извърши, адрес на данните, които ще се обработват.
126. 2) {106} Компютри с редуцирана система. Както и компютри със сложна система.
126. 3) {116} МИ с 1 и с 2 операнда

127. 162. Защо е полезно да бъде намален броят на АП в МИ?

127. 1) {105} За да може системата да работи по-бързо **адресацията да е по лесна**.
127. 2) {105} Полезно е за да може по-бързо да се извършват действията в МИ и за по-лесна работа.
127. 3) {106} За по-бързо изпълнение на МИ.
127. 4) {110} За да се оптимизира работата му. **[На кой му?]**
127. 5) {110} Защото времето за достигане до дадено АП е много по-голямо от времето, необходимо за четене/запис.
127. 6) {111, 113} Защото ЦП чете МИ и **колкото те са по-къси по-бързо ще е тяхното изпълнение**
127. 7) {115} за бързодействие и олекотяване на изпълнението
127. 8) {115} Намаляването на броя на адресните полета в МИ **увеличава бързодействието и изпълнението** на МИ.
127. 9) {121} Броят на адресните полета в машинните инструкции е полезно да бъде намален за **да се извършват по лесно МИ**.

127. 10) {125} Защото, ЦП чете МИ и колкото те са по-къси по-бързо ще е тяхното изпълнение.

128. 163. Как се елиминира четвъртото АП (за следваща МИ)?

128. 1) {105} Елиминирането на четвъртото АП за следващата машинна инструкция става чрез предаване на третата.
128. 2) {105} По подразбиране следващата машинна инструкция е на съседен адрес.
128. 3) {110} Като кода на следващата МИ се премести в резултата (3-тото АП)
128. 4) {111} Като машинните инструкции се разположат в последователни адреси
128. 5) {112} Като КОП за следващата МИ се запише на мястото на вече използвания операнд 1 (първото АП)
128. 6) {113} Като машините инструкции се разположат в последователни адреси.
128. 7) {122} Четвъртото АП се премахва, като информацията от него бива записана в третото АП.
128. 8) {123} Четвъртото адресно поле се елиминира като неговата стойност се зарежда директно в ЦП и се извиква при нейната необходимост
128. 9) {125} Като машинните инструкции се разположат в последователни адреси

129. 164. Какво представлява понятието „естествен ред на изпълнение на МП“?

129. 1) {102} Понятието „естествен ред на изпълнение на МП“ е изпълнение на действия на МП в последователен ред.
129. 2) {102} един компютър трябва да изпълнява зададеният му алгоритъм , явяващ се последователност от елементарна команда
129. 3) {105} При „естествен ред на изпълнение на МП“ МИ се разполагат в ОП по реда на тяхното изпълнение
129. 4) {111} Командата, следваща след текущата **[Всяка команда има две следващи след текущата! За коя от двете става въпрос?]** се намира на следващия адрес в паметта
129. 5) {113} Подразбиращи се МП.
129. 6) {115} Това е последователността на изпълнение на машини инструкции
129. 7) {121} Програмата се зарежда в ОП. От там се изпълняват нейните машинни инструкции използвайки входните данни. Всички междинни данни се записват в ОП. От дадените вх. данни и зададените от програмата преобразувания се получават изходните данни. {121} Цената на ОП намалява наполовина приблизително на всеки 3 години.
129. 8) {122} Командите **[Кои команди?]** са в последователни адреси в паметта.
129. 9) {123} Последователно изпълнение на действията
129. 10) {123} Естественият ред на изпълнение на МП е следният: ЦП прочита МИ, слес което прочита какво трябва да се използва за да изпълни МИ, след това запазва памет, която ще използва и накрая изпълнява МИ.
129. 11) {124} Естествената последователност на извършване на процесите и командите от МП.
129. 12) {126} след като се изпълни една МП, се изпълнява този, който е на следващия адрес в ОП.

130. 165. Как влияе елиминирането на АП за следваща МИ на конструирането на ЦП?

130. 1) {106} Как влияе – икономично за ОП. Елиминирането на АП за МИ означава, че при изпълнение на следващата инструкция Процесорът се обръща „естествено“ към следващия адрес на паметта.
130. 2) {113} Необходим е пренос на втория операнд по адреса на първия.
130. 3) {113} **В ЦП** се включва тривиалната операция за пренос на втория операнд по адреса, на първия операнд – резултат, тоест разширяване на машинният език с нов КОП
130. 4) {115} При премахване на АП се постига по-малък размер на МИ. Това предполага по-малък размер на регистрите и адресната шина, нужен за обработки на МИ.
130. 5) {117} Необходима е допълнителна операция пренос на втория операнд по адреса на първия операнд
130. 6) {122} Включва се тривиалната операция за пренос на втория операнд по адреса на първия
130. 7) {122} Нужен е програмен брояч – регистър помнещ текущата МИ
130. 8) {125} ЦП по-бързо ще прочете МИ.

131. 166. Как се елиминира третото АП (за резултат)?

131. 1) {104} Третото АП се елиминира **чрез операция за изтриване**
131. 2) {105} Елиминирането на адресното поле за резултат става, когато имаме цикличност на програмата или имаме метод, който да връща резултат дефиниран като отделен метод в програмата
131. 3) {114} Чрез адресирането по база с цел съкращение
131. 4) {115} Като резултата се държи в акумулатора
131. 5) {116} Ако стойността на операнда е необходима го местим с копиране на друг адре.

132. 167. Как влияе елиминирането на третото АП (за резултат) на конструирането на ЦП?

132. 1) {104} при елиминирането на третото АП резултатът се записва в I–и операнд. По този начин няма нужда от разхождане от едно място в ОП до друго и работата на ЦП става по-бърза
132. 2) {105} Резултатът се записва в регистри (или в паметта) т. е. необходими са регистри с общо предназначение.
132. 3) {111} Нужен е спец. регистър – програмен брояч, който да помни коя е текущата инструкция.
132. 4) {111} ЦП трябва да помни до къде е стигнало изпълнението в регистър, програмен брояч.
132. 5) {115} повишава производителността, увеличава изчислителна мощ
132. 6) {121} Виртуална машина, която аспенища асемблерния език.
132. 7) {122} Като помним резултата в адр1 Ако стойността на операнда е необходима го местим с копиране на друг адрес.
132. 8) {122} Не влияе по никакъв начин.

133. 168. Как се елиминира първото АП (за I операнд и резултат)?

133. 1) {105} В Централната памет се изработва регистър наречен акумулатор и се удвоява операцията за пренос
133. 2) {105} Първото АП се елиминира чрез използване на регистър в КОП.
133. 3) {106} Адресното поле се елиминира като информацията, която съдържа то под формата на битове (адресни клетки) се запаметява в ОП или пренася в следващите го АП
133. 4) {111} Използваме акумулатора
133. 5) {114} Първото АП се елиминира като първия операнд на (първото АП) се премести на второто АП, ако второто АП (операндна това място) е нужен се записва в акумулатор или ОП.

134. 169. Как влияе елиминирането първото АП (I операнд/резултат) на конструирането на ЦП?

134. 1) {110} Процесора става по-бърз, но има един недостатък – възможно е операнда да потрябва по-нататък в процеса, за това неговата стойност се запазва в регистър;
134. 2) {115} Елиминирането на първото АП влияе като **в ЦП се включва тривиалната операция** за пренос на втория операнд по адреса на първия резултат/операнд
134. 3) {124} Улеснява конструирането на процесора.
134. 4) {124} Намаля се времето за обработка

135. 170. Може ли една МИ да няма АП? Ако да – как, ако не – защо?

135. 1) {102} Не тъй като адресната част на МИ се поделя на отделни полета наречени адресни полета
135. 2) {104} Не. Няма как да бъде извикана за изпълнение.
135. 3) {105} Не защото машинната инструкция трябва да си има адресно поле за да бъде заредена в „ALU“ процесора.
135. 4) {113} Не може
135. 5) {115} Ако използваме още един акумулатор за втория операнд или ако използваме регистър УС, тогава зареждаме в стека ЬДР1, АДР2 и запазваме резултата пак там.

136. 172. Какво представлява понятието „адресност на машинна инструкция“ и кога се използва?

136. 1) {115} Адресност на МИ представлява съответното адресно поле, в което се записва МИ

137. 173. Какво е предназначението на ЦП?

137. 1) {102} Предназначението на ЦП е да управлява процесите в един компютър и компунентите му.
137. 2) {102} Обработка н постъпилата информация. по шината
137. 3) {102} ЦП е основната част във всяка компютърна система. Той извършва всички пресмятания в компютъра. В него се създават процесите.
137. 4) {104} Предназначението на ЦП е да подава информация към другите елементи на компютъра и да извършва пресмятанията в него.
137. 5) {105} Предназначението на централния процесор е да получи данни от оперативната памет, за да изчисли и върне резултат. За целта се ползва цикъла: Извличане – Декодиране – Резултат
137. 6) {105} Извършване на алгоритмично-логически действия
137. 7) {105} Централния процесор обработва информацията заредена в ОП
137. 8) {105} Централния процесор е един от главните компоненти в компютъра и неговото предназначение е **да изчислява** аритметно логически **изчисления**.
137. 9) {106} Централният процесор **извършва изчисление и операции**. Като скоростта му зависи от броя извършени операции за единица време.
137. 10) {111} Да изпълнява Машинни Инструкций
137. 11) {112} Централния процесор извършва всички необходими изчисления за работата на един компютър.
137. 12) {121} Централния процесор извършва изчисленията.
137. 13) {122} Да извършва указаните му операции/изчисления.
137. 14) {124} ЦП извършва всички пресмятания необх. в комп. Обработва данните както от паметта, така и от **входно/изходните устройства и периферията**.

138. 174. Какво представлява понятието „микропроцесор“?

138. 1) {105} Микропроцесорът е един от елементите на компютъра основни като включа още устройство за вход и изход, процесор, данна платка и памет.
138. 2) {106} това е процесор с по-малка памет, които е част от главния процесор, и извършва своите действия по-бавно.
138. 3) {110} Микропроцесор означава малък процесор, изграден от множество интегрални на 1 mm^2.
138. 4) {111} Това е **процес** на микрокомпютъра
138. 5) {113} част която обработва данни в формата на задачи
138. 6) {115, 122, 122, 123, 125} Процесор на микрокомпютър **[Не бива каруцата да се поставя пред коня!]**
138. 7) {121} Апарат за изчисление.
138. 8) {121} Микропроцесорът е умалена форма на ЦП. Той работи с по-малко операции и с по-малко (кеш cache)
138. 9) {123} Процесор, който изпълнява единствено и само предварително зададен алгоритъм.

139. 175. От какви части се състои един ЦП?

139. 1) {104} ЦП се състои от 3 части:
1) Информационен блок – използва се от ЦП да складира информация независима памет
2) Алгоритмичен блок – използва се да извършва математическите действия и задачи
3) Работен блок – за прочитане и обработване на програмата
139. 2) {104} При отношение главна подчинена програма подчин. трябва да извърши изцяло своята работа до възврат на главната
139. 3) {110} От блок управление, регистър, компаратор.

139. 4) {111} Тактов брояч. Кеш памет. Алгоритмичен блок – за смятане
139. 5) {114} Множество интегрални схеми
139. 6) {119} Управляващо устройство **[Твърде малко!]**
139. 7) {122} **Алгоритмо** логическо устройство, програмен брояч
139. 8) {124} ЦП се състои от ядро, три вида кеш памет
139. 9) {124} Тактов генератор, генериращ тактовете на работа, 102схеми Конструиран е от интегрални схеми с високо ниво на интеграция.
139. 10) {125} Ядра, кеш памет

140. 176. Какво е предназначението на УУ на ЦП?

140. 1) {106} Управляващото устройство **задвижва механизма на работа на централната памет**
140. 2) {111} Предназначението на УУ на ЦП е **да разбере съответният машинен език** и подготви за изпълнение машинната програма.
140. 3) {112} Коя МИ да се изпълни и кога.
140. 4) {114} УУ използват данните от ОП за изпълнението на МИ.
140. 5) {114} УУ на ЦП задава задачите с които трябва да се справи АЛУ.
140. 6) {115} УУ управлява АЛУ (аритметико логическо устройство) и РЗ (регистров блок) като чрез тях изпълнява алгоритъма на програмата извлечи–декодирай–изпълни.
140. 7) {121} П-е да приеме постъпилите МИ да ги декодира и изпрати на АЛУ за изпълнение.
140. 8) {122} Да управлява АЛУ и РБ, чрез тях изпълнява алгоритъма на програмата. Декодиране, изпълнение и извличане.
140. 9) {124} УУ (управляващо устройство) управлява АЛУ (аритметико-логическото у-во) и РБ (регистров блок)
140. 10) {125} Да управлява процесите.
140. 11) {125} УУ управлява АЛУ и РБ.
140. 12) {126} Определя кои операции да извърши ЦП и в какъв ред.

141. 177. Какво е предназначението на АЛУ на ЦП?

141. 1) {102} Адресната лента преразпределя кое къде и в какво последователност ще се запишат при възпроизведата процесите по четене и запис (read 8)
141. 2) {121} Когато заявка за прекъсване, изпратена от външно устройство (към процесора) (с цел да се обърне внимание на входно/изходната част, ако тази заявка има маска на прекъсване, то тя не може да бъде отлагана.

142. 178. Какво представлява регистровият блок на ЦП и за какво е предназначен той?

142. 1) {115} Регистър за временно съхраняване на данни в ЦП
142. 2) {119} **Съдържа** всички основни **операции** са предназначени за по-бърз достъп.

143. 179. Какви видове регистри има в състава на ЦП?

143. 1) {112} И – регистър на инструкциите, А – акумолаторен, Х – индексен, РАП – регистър за адрес ат Паметта, РУ – регистър на условията ПБ – програмен брояч, и РОП – регистри с общо предназначение.
143. 2) {114} Въпреки сложността, размерът устройството и общия вид на процесорите се променят своите механизми за да определи дали има съвпадение на даден адрес вътре в регистрите си
143. 3) {115} Акумулаторен, регистър на стек, индексен
143. 4) {121} Динамично модифициране на адрес.
143. 5) {121} Адресни регистри, регистри на управление
143. 6) {122} Еднопроцесорни регистри и паралелно процедурни регистри
143. 7) {123} Р
– регистър условия – РУ
– Програмен броям – ПБ

– Регистър инструкции – РИ
– Регистър адресни – РАП

143. 8) {123} Регистър на действията, регистър на паметта, регистър на операциите Регистър на периферията
143. 9) {124} 1) адресен регистър 2)Регистър на условията 3) служебен регистър
143. 10) {124} За вход, изход и резултат

144. 180. Какво е предназначението на служебните регистри на ЦП?

144. 1) {105} Да посочат различните събития при изчислението например препълване знак,
144. 2) {105} Служебните регистри достъпват инструкциите за работа на УУ на ЦП.
144. 3) {111} глобален достъп до данни
144. 4) {111} Служат за запомняне (съхранение) на данните.
144. 5) {113} Да разбире зададената му програма и да я изпълни
144. 6) {121} Това са регистри, които ЦП може да използва при …
144. 7) {122} Предназначението на служебните регистри е запомняне и съхранение на данни
144. 8) {122} Предназначението на служебните регистри на ЦП е **те да изпълняват програмата**. В тях се записват и съхраняват данни за МИ адрес на данни в ОП
144. 9) {122} Регистри които могат да се ползват от служителите.
144. 10) {125} В тях се записват междинни данни и адреси на МИ

145. 181. Посочете служебните регистри (поне 3) на ЦП.

145. 1) {104} Служебните регистри на ЦП са **запомняне, обработване и предаване** на информацията
145. 2) {126} Регистър на адресите, програмен брояч, регистър на операндите.

146. 182. Какво е предназначението на програмния брояч на ЦП?

146. 1) {101, 122} В него се съхранява информация за това до къде е стигнало изпълнението на МИ.
146. 2) {103} Предназначението на програмния брояч ци посочва размера на обработваните данни и постъпването на машинна инструкция
146. 3) {105} Регистър в ЦП
146. 4) {105} В него се съхранява информацията за това **докъде е стигнало изпълнението на МИ**.
146. 5) {110} Да помни изпълняваната инструкция
146. 6) {110} да изчислява промените в ЦП
146. 7) {111, 113} Да следва коя инструкция се изпълнява в момента
146. 8) {111, 112} да се следи коя инструкция да се изпълнява в момента
146. 9) {112} Да следи коя инструкция се изпълнява в момента.
146. 10) {112} Отброява времето за изпълнение на МИ.
146. 11) {113} След изпълнението на МИ се увеличава с 1
146. 12) {114} В него се съхранява информацията докъде е стигнало изчислението на МИ.
146. 13) {114} Програмния
146. 14) {115} Съхранява адреса на инструкцията и го увеличава с 1
146. 15) {120} Програмния брояч **изброява действията**, които ЦП извършва за дадена МИ **и преглежда действията** за да съкрати ненужните операции на ЦП.
146. 16) {121} Да следи кода
146. 17) {122} Програмния брояч на ЦП е предназначен да брои процеси които се извършват.
146. 18) {122} Да следи коя инструкция се изпълнява в момента
146. 19) {124} Да следи коя инструкция се изпълнява в момента.
146. 20) {124} да отброява използваната памет
146. 21) {124} Да разбере задачата под формата на алгоритмичен код и да я изпълни.
146. 22) {125, 125} Предназначението на програмния брояч на ЦП е да следи коя инструкция се изпълнява в момента.
146. 23)

147. 183. Какво е предназначението на регистъра на инструкциите на ЦП?

147. 1) {103} Там се пазят всички инструкции на централния процесор.
147. 2) {105} Съхранява адресите на инструкциите.
147. 3) {110} РУ следи за изпълнението на машинните инструкции (МИ) в централния процесор
147. 4) {111} Съхранява инструкциите де работа на централния процесор.
147. 5) {113} Временна памет – за временно съхраняване на данни в процесора вместо в паметта
147. 6) {115} Да регистрира подадените инструкции
147. 7) {116} Участва във фаза „извличане на МИ“ чрез даннова шина.
147. 8) {121} Използва се като мост между ЦП с ОП.
147. 9) {125} спомага за правилното изпълнение на програмата
147. 10) {126} Да бъдат използвани от програмиста за негови цели.

148. 184. Какво е предназначението на регистъра за адрес от паметта на ЦП?

148. 1) {102} В регистъра за адрес от паметта на централния процесор се съхраняват адресите на най-често използваните машинни инструкции
148. 2) {109} Регистъра за адрес от паметта на ЦП служи за определяне на различните процеси и тяхното изпълнение
148. 3) {112} Предназначението на регистъра за адрес от паметта на ЦП е да определя адреса за паметта.
148. 4) {114} съдържа адресите от оперантите на изпълняваната програма.
148. 5) {114} Адресация за бърз достъп до **елементите от паметта на ЦП**.
148. 6) {123} Запазват се адресите на паметта в регистъра. Улеснява се достъпът до данните.
148. 7) {124} Да заделя и посочва данни записани в паметта на ЦП.
148. 8) {124} Предназначението му е да запазва адреса в паметта, където се намира следващата машинна инструкция.
148. 9) {125} Те служат на програмата за изпълнение на необходимите ѝ операции чрез тях и в тях.
148. 10) {125} Съхранява изпълнявания адрес на данните в ОП.

149. 185. Какво е предназначението на програмно достъпните регистри на ЦП?

149. 1) {111, 113} съхраняват междинните данни на изпълняваната МИ.
149. 2) {115} **да изпълняват програми** – да са доствини до програмата
149. 3) {118} Индексен, указател на текст и акумулатор
149. 4) {121} Наличието на информация няма при извършване на различни процеси.
149. 5) {121} Използват се за съхранения на данни и адреси ЦП работи по бързо с тях. Могат да бъдат адресирани чрез езиците от ниско ниво.
149. 6) {123} Целта е дадена програма да получи директен достъп до данните в тях (пример – отчет на съхраняваната информация в регистрите)
149. 7) {125} Да разбере зададената програма и да я изпълни
149. 8) {125} Ускорява работата на Ц. П.
149. 9) {126} Служат на програмата за извършване на необходимите ѝ данни чрез тях в тях
149. 10) {126} Ефективното поле се определя от и няколко регистъра.

150. 186. За какво служат регистрите от тип акумулатор на ЦП и защо са въведени?

150. 1) {107} Регистър на онструкциите
Регистър за адрес от паметта
150. 2) {107} За **натриване** на данни за да не се прехвърлят постоянно данните от ЦП към ОП и обратното
150. 3) {122} Служат за улесняване работата на ЦП Въведени са за подобряване на работата и бързодействието на ЦП.
150. 4) {125} Те са съставени от два акумолатора Регистрите от акумолатор на ЦП са с уникални части.

151. 188. За какво служат регистрите от тип указател на стек на ЦП?

151. 1) {116} указват местоположението на стека

152. 189. Посочете основните програмно достъпни регистри на ЦП.

152. 1) {123} Регистри до оперативната памет.
152. 2) {124} AX, BX, CX
152. 3) {124} акумулаторни, базови, индексни
152. 4) {125} Регистрите с общо предназначение (Акумулаторни, индексни и указател на стека)

153. 190. Какво представляват акумулаторните процесори?

153. 1) {105} Акумулаторните процесори издържат по-дълго време
153. 2) {111} Те са енергонезанисими Съхраняват информацията за астрономическото време и др. Намират се на отделен чип.
153. 3) {121} Акумолаторните процесори, са тези които в настоящето се използват в компютрите. При тях информацията се представя под формата на поредица от „0“ и „1“ **[А при другите като поредица от фурки и вретена!]**
153. 4) {124} и конструира част от програмата.
153. 5) {125} При тях регистрите не се посочват, инструкциите са къси и не се мисли кога, как и кои ще се използват.

154. 191. Каква е идеята на процесорите с регистри с общо предназначение?

154. 1) {105} Изръбнямат различни видове задачи използват се от масовия потребител (домашните компютри) могат да се използват за по-толям брой прилочения.
154. 2) {105} Идеята е да се използва по-малка част от паметта, да бъде по подредено и стуктурирано, и процесорът да може да осъществява своите действия по-бързо.
154. 3) {111} да работят с разбични типова данни
154. 4) {113} Да регистрира процесите в паметта.

155. 192. Какво представляват процесорите с регистри с общо предназначение?

155. 1) {110} Това са процесори, които могат да изпълняват различен набор от функций.
155. 2) {114} Такива процесори не се използват за решаване на сложни диференциални уравнения

156. 193. Каква е идеята на стековите процесори?

156. 1) {105} Стековите процесори се създават, ако процесорът не е изпълнява автоувеличение и автонамаление, за да може да работи със стек, това се решава при конструирането.
156. 2) {105} Обработка на голямо количество информация за минимално време
156. 3) {106} Идеята на стековите процесори и по-бързото изпълнение на програмите.
156. 4) {106} Те не използват регистри. Записват резултата на място.
156. 5) {111} Вместо регистри за данни, използват стек, па резултатът винаги се записва на върха на стека.
156. 6) {113} Стековите процесори са реализирани посредством стек, като при тях идеята е последния влязъм – първи излиза
156. 7) {114} Аритметични пресмятания
156. 8) {114} Те нямат регистри за данни. Използват, стек за данни да заемат операндите на върха на резултата
156. 9) {124} Те нареждат зададените им команди в стек, и като ги изпълнят след това 1 по 1 а не няколко наведнъж, спестяват време.
156. 10) {124} Работа със системната памет.

157. 194. Къде се използват стекови процесори?

157. 1) {111} Където няма нужда от процесори с регистри за данни.
157. 2) {111} Вместо регистри използват стек като **резултата се излива** най-отгоре на стека

157. 3) {111} не в мето регистрация на даните използват смет за даните а резултат виран записва наверх на стек
157. 4) {113} При обхождане на (големи) масиви.
157. 5) {116} За аритметични пресмятания например
157. 6) {121} – стекови процесори се използват при съвременните компютри (те са цифрови, двоични)

158. 195A. Посочете поне две предимства на акумулаторните процесори в сравнение с процесорите с регистри с общо предназначение.

158. 1) {106} По бърз, по-малко регистри.
158. 2) {114} По бърз достъп до данните, няма стек
158. 3) {121} Те са по-мощни и по-надеждни
158. 4) {121} Бърз достъп в акумулаторите, недостатъците са малък брой параметри.
158. 5) {121} По.бързи са
158. 6) {125} Предимства по-лесни за програмиране тъй като не трябва да указваме в кой регистър да записваме Недостатъци – ако трябва да ползваме повече от 1 регистър тогава ЦП с акумолатори са по-трудно програмуени.

159. 195C. Посочете поне две предимства на процесорите с регистри с общо предназначение в сравнение с акумулаторните процесори.

159. 1) {104} По-бърз достъп и улеснено използване на ресурсите
159. 2) {105} Позволява да се ползват произволно от програмист взаимнозаменяеми, по-малко специални машинни инструкции
159. 3) {115} Повече свободна памет
– не се налага
159. 4) {115} Регистри с общо предназначение – универсални регистри с достатъчен брой ускорени изпълнения трябва да се посочват чрез номерата си, трябва да се мисли за разпределението им. **[Подобен отговор показва, че студентът не може да различи зърното от плявата!]**

160. 196. От какви фази се състои изпълнението на една МИ?

160. 1) {105} операд 1 (операнд 2) запис / и има още 1.
160. 2) {106} Вход или задаване на данни, обработка, изпълнение (измисление) и запаметяване на получения резултат
160. 3) {123} Четене, обработване, изрълнение

161. 198. Защо е необходима фаза „декодиране на МИ“?

161. 1) {120} МИ се нуждаят от декодиране понеже те се пишат на програмен език от програмиста, а после машината ги превръща в комбинация от 0 и 1 които тя разбира.

162. 199. Какво извършва ЦП по време на фаза „изпълнение на МИ“ при обработващите МИ?

162. 1) {105} ЦП прочита МИ и я изпълнява. Той трябва да знае до къде е стигнало изпълнението на МИ чрез програмен брояч.
162. 2) {106} ЦП следва инструкциите и посредством адресната, данновата и управляващата шина, след това изпълнение на МИ, **обработената инструкция се записва (запаметява) в паметта**.
162. 3) {110} **Прочита** и изпълнява **машинните инструкции**
162. 4) {111} Обработва данните и съхранява резултата.
162. 5) {111} ЦП изпраща към ОП при изпълнение на дадена машинна инструкция. Чрез него **[Т. е. чрез ЦП!]** при обработващите Ми ЦП чете или записва данни, а при управляващите Ми се определя следващата Ми.
162. 6) {115} Операции с един операнд
162. 7) {121} Извършва проверка на достъпните регистри и проверява данни

162. 8) {124} Извършва командите зададени в определената машинна инструкция
162. 9) {125} след като изпълним една МИ се изпълнява тази, е на следващия адрес в ОП.
162. 10) {125} Обработват данните, тоест извършват нужните изчисление

163. 200. Защо се говори за цикъл „извлечи-декодирай-изпълни“ на УУ на ЦП?

163. 1) {101} Защото управляващото устройство извършва този цикъл при изпълнението на поредната машинна инструкция.
163. 2) {111} Изпълнението на една Ц. И. се състои от три фази.
Извличане, Дешифриране и изпълнение та машинна инструкция
Реализиране на фаза извличане М. И.
1. Адресна шина = програмен брояч.
2. Сигнал четене/запис = Четене
3. Програмният брояч се увеличава с 1.
4. Регистър на инструкциите = Даннова шина.
163. 3) {118} Това са трите фази на изпълнение на една машинна инструкция, извършвано от упр. устройство на централния процесор
163. 4) {121} Защото това е схемата по която, процеси извършени през УУ да бъдат **усъществени** от ЦП.
163. 5) {122} Говори се за цикъл „извлечи–декодирай–изпълни“ на УУ, защото то „извлича“ информацията от паметта на КС, „декодира“ я и „изпълнява“ програмата.
163. 6) {124} упра. у-во извършва този цикъл при изпълнението на поредната машинна и-ция.
163. 7) {124} Защото записите са кодирани, трябва се извлекът от стека да се декодират и след това да се пълнят.
163. 8) {125} Защото управляващото устройство извършва този цикъл при изпълнението на порезна машинна инструкция. .
163. 9) {126} Защото управлението на устройство изнършва в този цикъл при изпълнението на поредната машинна инструкция

164. 201. Как ЦП „разбира“ къде в ОП има МИ и къде – данни?

164. 1) {114} Никак. Там където сочи ПБ, там е инструкция Но не е сигурно дали са инструкции или данни
164. 2) {115} зависи как е адресирана думата
164. 3) {116} посредством регистъра на данни в оперативната памет
164. 4) {124} Като ги разпознава и разграничава
164. 5) {125} Ако е в фаза изпълни думата се тълкува като данни.

165. 201. Как ЦП „разбира“ къде в ОП има МИ и къде – данни?

165. 1) {105} адреса на клетките
165. 2) {111} Зависи в какво състояние се намира ЦП. Ако е четене се възприемат като данни, в противен случай **[Т. е. запис!]** като МИ.
165. 3) {111} с данновата шина – тя дава данни, а МИ по адресната шина

166. 202. Какви са особеностите на фаза „изпълнение на МИ“ при управляващите МИ?

166. 1) {114} зареждащата програма е програма която зарежда главната оперативна памет на Асемблер.
166. 2) {116} Посредством РАП от ОП се четат операндите АЛУ извършва необходимите изчисления Посредством РАП **от ОП се записва** резултата
166. 3) {121} От ОП посредством регистъра за адрес от паметта се извличат операндите, аритметико-богическото устройство (АЛУ) извършва необходимите изчисления и отново чрез регистъра за адрес от паметта се записват резултатите в ОП (оперативната памет).
166. 4) {121} на базата на КОП се разбира какво действие да се извърши и се изпълнява.
166. 5) {121} Управляващите МИ Не променят регистъра на условията. Чрез тях се извършва преход към друга инструкция ири наличие на определено условие

166. 6) {122} извършват се изчисление и резултатът се изпраща към ОП
166. 7) {124} Посредством РАП от ОП се четат операциите:
АЛУ извършва необходимите изчисления. Посредством РАП от ОП се записва резултат

167. 203. Какви видове ЦП съществуват въз основа на това как УУ осъществява фаза „изпълнение на МИ“?

167. 1) {105} с регистри, пряка.
167. 2) {110} Матричен
167. 3) {115} От ОП се четат операндите и се прехвърлят в РАП

168. 204. Как реализират фаза „изпълнение на МИ“ апаратните ЦП?

168. 1) {105} Последователно. Прочита операнди, прочита код на машинна инструкция, изпълнява МИ

169. 205. Какво представляват „микропрограмируемите ЦП“?

169. 1) {106} Микропрограмируемите ЦП са процесори с общо предназначение най-често. Инструкциите които изпълняват се управляват от програмата (програмите) иначе казано software (наместо превключване на електрически вериги и устройства), както това е било в началото на компютърната техника,
169. 2) {110} Процесори които се използват за точно определена задача.

170. 206. Каква е връзката между понятията микропроцесор и микропрограмируем процесор?

170. 1) {102} Микро програмируемите процесори – чрез тях се следва избрана микропрограма, а самият микропроцесор е универсален. Сменим ли ЦП, губим всички създадени от него микропрограми
170. 2) {103} Микропрограмируемите процесори са частен случай на микропроцесор. Микропрограмируемите работат с микропрограми реализиращи фазите: извлечи, декодирай, изпълни.
170. 3) {105} И двата изпълняват код от Машинен Език (МЕ)
170. 4) {105} Връзката между понятията микропроцесор и микропрограмируем процесор, е че чикропроцесор може да бъде програмиран на микроравнище
170. 5) {107} Процесор на микрокомпютър.
170. 6) {111} Микропрограмируемия процесор е частен случай микропроцесора.
170. 7) {112} Еднакви са, само че микропрограмируемият процесор може да се програмира от програмиста.
170. 8) {113} Може да се програмира многократно и да се изпълнява МИ. Микропрограмируем процесор
Микропроцесор – може да се програмира и да изпълнява само една програма
нвапример програматор за печка, пералня и др.
170. 9) {121} Микропрограмируемите процесориса частенслучай на микропроцесорите.
170. 10) {123} Микропрограмируем процесор е микрапрацесор, който може да бъде програмиран.
170. 11) {123} Микропроцесора директно изпълнява дадената МИ докато Микропрограмируемият процесор може да изпълни допълнителни МИ пре.девинирани преди това и свързани с първичната МИ
170. 12) {124} Микропрограмируемите процесори са частен случай на реализиращи фазите: извлечи, декодирай, изпълни

171. 207А. Защо МЕ на всички ЦП не са еднакви?

171. 1) {105} Защото машините езици са различни при различните процесори.
171. 2) {107} Машинните език на всички централни процесори е различен, защото за всеки централен процесор има индивидуален машинен език.
171. 3) {109} Защото не всички ЦП извършват едни и същи операции

171. 4) {112} **МЕ** на всички ЦП **са еднакви** за да може всеки ЦП да чете данните използвани и записвани от друг ЦП.
171. 5) {112} Всички ЦП имат различни типове инструкции.
171. 6) {112} Зависи от производителността и от годината.
171. 7) {121} Защото всеки ЦП е различно конструиран и произведен от различни компоненти.

172. 207В. Еднакви ли са МЕ на всички ЦП?

172. 1) {111} Да, МЕ на всички ЦП са еднакви.
172. 2) {114} Да, еднакви са.

173. 208. По какво основно се различават МЕ?

173. 1) {102} МЕ се различава по това каква машина управляват и по начина на изпълнение.
173. 2) {106} По-размера.
173. 3) {106} МЕ се различават по Ц.П.
173. 4) {114} По равнището на програмиране
173. 5) {115} МЕ на различните ЦП са винаги различни, поради особеностите на всеки ЦП.
173. 6) {122} различните МЕ имат различни Асемблери
173. 7) {124} Различават се по бързодействие, начин на работе, време на възникване.
173. 8) {124} Машинните езици се различават по това дали са на високо или ниско равнище. Има разлика при начина на писане на функции и команди.

174. 209А. До какво води следствието от различието на МЕ?

174. 1) {111} **Микропрограмата** на един ЦП е неразбираема за друг ЦП
174. 2) {111, 111} **Микропрограма** на един ЦП е абсолютно неразбираема за друг ЦП
174. 3) {119} По бавно адресиране от ЦП.
174. 4) {121} Различието на Машинният Език Води до проблеми който трябва да се решават само с програмиране на ново. на даденият елемент който се различава. проблеми при свързване на 2 различни МЕ.
174. 5) {121} Поради различието в машинните езици, понякога се получава невъзможност за прочитане на информацията записана на един компютър от друг
174. 6) {122} **Микропрограма** на един ЦП е абсолютно неразбираема за друг централен процесор
174. 7) {123} Води до появата на различни видове транслатори.
174. 8) {124} Програмата спира да работи. Всеки МЕ е индивидуален за всеки ЦП. Защото се изработва от различни хора и без точно определени критерии.

175. 209В. Какво е следствието от различието на МЕ?

175. 1) {111} **Микропрограмата** на един ЦП е абсолютно неразбираема за друг ЦП
175. 2) {111} Трудно писане на програми. Трудно четене. Използват се всички ресурси на ЦП.
175. 3) {111} **Микропрограмата** на един ЦП е неразбираема за друг ЦП
175. 4) {121} Следствието е че всеки процесор има собствен МЕ и при промяна на проце3сора се губят всички негови **подпрограми**
175. 5) {126} МП на един ЦП е неразбираема за другите ЦП, това означава, че сменим ли ЦП губим всички създадени за него МП
175. 6) {126} Предимството от RISK – опростяване на ЦП (схема за работа с МИ) и работи по бързо Стандартни инструкций малко на брой

176. 210. По какви пътища може да бъде елиминиран недостатъкът от различието на МЕ на два ЦП за да могат потребителите да продължат да използват създадените МП?

176. 1) {102} Като се използва една и съща операционна система, която превежда различните машинни езици на един език или се използва една и съща адресна регистрация

177. 211B. Посочете предимства на програмната емулация като подход за елиминиране на недостатъка от различието на два МЕ.

177. 1) {115} Свързано е с legacy техника, се и с това, че всеки процесор има собствен МЕ. Използва се за използваеим програми писани за други ЦП или по-стари модели на същия.

178. 211D. Посочете недостатъци на програмната емулация като подход за елиминиране на недостатъка от различието на два МЕ.

178. 1) Загуба на точност заради различно представяне на типовете данни.

179. 212. Каква е идеята на апаратната съвместимост отдолу нагоре?

179. 1) {104} Идеята на апаратната съвместимост е да бъдат свързани по между си отделните компоненти на съответната програма
179. 2) {105} Съвместимост е необходима за да се използват всички съществуващи до момента програми
179. 3) {105} Апаратната съвместимост отдолу нагоре се използва при стек. Идеята е на върха на стека на мястото на операндите да се запишат резултатите.
179. 4) {105} Апаратната съвместимост отдолу нагоре позволява МП на по-стар ЦП да бъде изпълнявана на ЦП от по-ново поколение. Пример МП за ЦП 80286 може да бъде изпълнена на ЦП 80468 на Intel
179. 5) {111} При апаратната съвместимост по-новите процесори поддържат (интегрират) МИ на предхождащите ги процесори и интегрират нови МИ **[Този отговор е много мътен за да бъде признат за верен!]**
179. 6) {111} Всеки следващ процесор де надгражда предхождащите го
179. 7) {113} Мести само при уникални решенира да има малко регистър с общо предназначение уникални
179. 8) {114} МИ се създава като съществуваща и се допълва с нови МИ
179. 9) {122} Прави така, че всички апарати да бъдат съвместими
179. 10) {123} За да може най напред апаратите да бъдат съвместими на асемблерно ниво.
179. 11) {126} По-новите модели да запазят функционалността на по-старите модели.

180. 213. Каква е идеята на програмната емулация?

180. 1) {103} Приспособяването на програма към дадена среда за изпълнение.
180. 2) {105} Има за цел програмата да бъде разчетена от машинния език
180. 3) {111} Цикълът извлечи-декодирай-изпълни има **алгометричен** характер и **може да се управлява програмно**
180. 4) {113} Стартиране на програма Транслацията ѝ.

181. 215. Какви видове МЕ съществуват по отношение на тяхната съвкупност от МИ?

181. 1) {102} Специализирани и универсални.

182. 219. Каква е връзката между МЕ (CISC или RISC) и начина на изработка на УУ на ЦП?

182. 1) {111, 114} Те са тясно свързани, тъй като **с помощта на машинния език** управляващото устройство на ЦП **организира изпълнението на машинната програма**.В зависимост от МЕ УУ се изработва по различен начин. При CISC цикълът на изпълнение на МИ на УУ има ниско янго. характер и може да се регулира по програмен път от ЦП.
182. 2) {124} При RISC УУ се конструира според факта, че всички инструкции имат еднаква структура, а при CISC структурата е различна
182. 3) {124} Колкото е по опростен машинния език, толкова по-лесен е начина на изработка на управляващото устройство на процесора.
182. 4) {125} Те са тясно свързани, тъй като с помощта на МЕ управляващото устройство на ЦП организира изпълнението на машинната програма. При CISC списъкът на изпълнение на МИ на УУ има чисто алторитмичен характер и може да се регулира ат ЦП по програмен път.

182. 5) {125} Различните „семейства“ централни процесори (ЦП) използватразлични машинни езици (МЕ). По новите поколения използват МЕ на предшественика си, но с някои разширения.
182. 6) {126} Те са тясно свързани, тъй като с помощта на машинния език управляващото устройство на ЦП организира изпълнението на машинната програма.

183. 220. Как може да бъде повишена производителността над границите, продиктувани от използваните елементи и технологии?

183. 1) {105} рез използване на копроцесори.
183. 2) {114} като се **овиличи** честотата на тактовия генератор.
183. 3) {115} Чрез по-толямо насищане ня интегралната схема с повече на брой и ва по-малки елементи, т. е. постигане на по-голяма производителност на дадена план. (с по-висока степен на интеграция)
183. 4) {121} Като се подобри системата

184. 221. Посочете най-елементарния начин за въвеждане на паралелизъм в работата на ЦП.

184. 1) {111} Най-елементарният начин е да припокрием етап „изпълнение“ с етап „Извличане“.
184. 2) {111} Най-простият начин е краят на една МИ и началото на друга.
184. 3) {112} Паралелното управление на 2 ЦП чрез т. нар. „паралелно програмиране“
184. 4) {114} Операция извличане и записване да се изпълняват заедно (да се застъпват).
184. 5) {115} Работа с повече от един процесор.
184. 6) {119} Като се изпълняват **последователно** части от различни процеси
184. 7) {122} Възможност за обработка на машинните инструкции от повече от един процесор. Характерно е за суперкомпютрите
184. 8) {122} Чрез копрограми.
184. 9) {124} Край и начало. Краят на една МИ и началото на друга

185. 222. Какво представляват „конвейерните процесори“?

185. 1) {101} При тях изпълнението на МИ има три фази от които последната е от две стъпка

186. 231. Какво представлява понятието „кеш-памет“?

186. 1) {110} Високо скоростна памет. В нея се записват инструкциите, които ще се ползват.
186. 2) {114} съхранение на специфична информация

187. 232. Каква е основната идея на „кеш-паметта“?

187. 1) {104} Играе роля на буфер между ЦП и ОП.
187. 2) {105} Служи за буфер на централния процесор и оперативната памет
187. 3) {105} Кеш-паметта се явява буфер между оперативната памет и централният процесор. Тя е полезна, когато те работят с различна скорост. Ако ЦП и ОП работят с еднаква скорост, кеш-паметта е излишна.
187. 4) {105} Кеш паметта е процесорна и служи за огледално копие на част или цяла програма. Идеята е да създаде бързо копие/увеличаване на скоростта.
187. 5) {106} Да спомогне за повишаването на **производителността на МИ в ЦП**.
187. 6) {115} Играе ролята на буфер между централния процесор и оперативната памет
187. 7) {115} Играе ролята на буфер м/у бавно ОП и бързия ЦП
187. 8) {116} Кеш-паметта играе ролята на буфер между бавната ОП и бързия ЦП.
187. 9) {116} Тя е заложена в съществуването на алгоритми, според които голяма по обем бавна памет и малка по обем бърза памет могат да работят почти със скоростта на бързата памет. Тази малка по обем бърза памет служи за буфер между ЦП и ОП и се нарича кеш-памет.

188. 234А. Кога е полезно да имаме външна „кеш-памет“?

188. 1) {111} Когато ОП е бавна.
188. 2) {111} Когато ни е нужен буфер между ЦП и ОП

188. 3) {113} Когато процесора е претоварен с много заявки за изчисление
188. 4) {121} Полезно е когато ОП е обикновена (бавна)
188. 5) {121} CD, DVD.
188. 6) {123} Винаги когато ни трябва по голямо бързодействие, по-висока производителност Безсмислено е когато ОП е много бърза
188. 7) {123} Полезно е да има кеш-памет (външна) по всяко време дори да е само за обикновен back up
188. 8) {126} Полезно е ако ОП е обикновена, но ако цялата ОП е бърза тогава няма смисъл.

189. 234В. Кога е безсмислено да имаме външна „кеш-памет“?

189. 1) {111} Когато машината извършва само едни и същи операции
189. 2) {111} Когато цялата ОП е бърза
189. 3) {114} Полезно е, ако ОП е обикновена (бавна).
189. 4) {121} Когато ОП работи бързо спрямо процесора и вътрешната кеш-памет (по-бърза) е достатъчна.
189. 5) {122} Безсмислено е, когато цялата ОП е бърза.
189. 6) {122} Когато операционната система е достатъчно бърза, или не ни трябва повече бързина или производителност.
189. 7) {124} Когато използваме информацията редивно не е нужна кеш-паметта.
189. 8) {125} Полезно е, ако ОП е обикновена (бавна) ако цябата ни ОП е бърза е безсмислено.

190. 235. Защо някои ЦП имат два комплекта „кеш-памет”?

190. 1) {105} Едната кеш памет служи за един евентуален буфер
190. 2) {105} За по-бързия пренос на данни, които увеличават производителността на компютъра.
190. 3) {110} За да бъде по-бързо изпълнението на МИ-и.
190. 4) {114} едната част обработва данни, а другата е за инструкции
190. 5) {115} **За** данни и **заявки**
190. 6) {115} Два типа CASH MEMORY – увеличаване скоростта на обработка на данните м/у ОП и ЦП. Необходима, поради различната скорост на процесора
190. 7) {124} За да може уа нещо се случи с единият. Да може да Възстановят процесите от втората кеш памет.
190. 8) {124} Едната е за данни, другата е за инструкции.
190. 9) {124} Всички съвременни процесори имат поне два вида кеш-памет L1 и L2. L1 е вътре в кристала на ЦП и L2 на платката на процесора. Те позволяват на ЦП да обменят данни с ОП много по-бързо.
190. 10) {125} За по-бърза работа в едната се пазят инструкциите, а в другата временните данни с които работят те

191. 237. Какво представлява вътрешната „кеш-памет“?

191. 1) {106} Представлява паметта, която е „кеширана“ от процесора

192. 238. Какво представлява външната „кеш-памет“?

192. 1) {105} Тя е буфер между оперативната памет и ЦП.

193. 240А. Защо малкият брой жици в адресната шина е грешка на конструкторите на ЦП?

193. 1) {103} Защото малкият брой жици не осигурява бързодействие и голям обмен на данни за малко време
193. 2) {111} Защото Адресното поле трябва да е по-голямо и се нарушава еднотипната обработка на данни, които са в клетки от ОП.
193. 3) {111} Могат да пренасят по-малко информация
193. 4) {124} Така се забавя работата на ЦП.
193. 5) {124} Защото са малко и не може да ги разпознае.

194. 240В. Каква е най-често срещаната грешка на конструкторите на ЦП?

194. 1) {104} Несъответствието между скоростта на пренос на данни на шината и скоростта на обработване на данните от ЦП
194. 2) {105} Най-често грешка е подаване на грешни параметри.
194. 3) {111} Малкият адрес в адресните полета.
194. 4) {111} Малкото адресно поле, защото той определя максималния обем ОП.
194. 5) {111} Използване на стойност вместо адрес и обратно
194. 6) {113} **Малкото** (късо) **адресно поле**, защото той определя максималния обем на ОП
194. 7) {114} Малко **наднесно** поле
194. 8) {114} Малко (късо адресно поле, защото той определя максималния обем ПО
194. 9) {122} Централния процесор е предназначен да разпознае и да изпълни зададената ву програма. Понякога при конструкторите възникват грешки, когато програмата не може да бъде разпозната и изпълнена от него.
194. 10) {125} Малкото адресно поле, което от своя страна определя количеството ОП.
194. 11) {125} Малкото (късо) адресно поле, защото той определя максималния обем ОП.

195. 241. Какви проблеми създава поставянето на пълен адрес от ОП във всяко АП?

195. 1) {113} Проблеми се състоят в това, че разполагането може да стане по два начина старшия битов на думата могат да бъдат в клетките с по-малък адрес (голямокраен компютър или погулям адрес (малокраен компютър)
195. 2) {121} Програмата става по-голяма (заради по-дългите машинни инструкции – съдържат пълни адреси), поради същата причина извличането на машинни инструкции от ОП отнема повече време.
195. 3) {121} По-бавна работа на ЦП.
195. 4) {121} Не
195. 5) {122} Един за инструкциите и един за данните. С цел бързина.
195. 6) {125} Пълният адрес увеличава времето за търсене и изразява повече ресурсна мощ

196. 242. Какво представлява понятието „способ за адресиране“?

196. 1) {104} Способ за адресиране означава метода който използвахме и избрахме за адресирането
196. 2) {104} Способ за адресиране се използва за да се даде адрес на даден файл за да може той да бъде преместен.
196. 3) {105} Възможността чрез адреси в паметта да определяме мястото на операндите.
196. 4) {106} Понятието „способ за адресиране“ се използва когато имаме възможност да елиминираме първото адресно поле
196. 5) {113} Видове адресация и способ за адресиране са синоними.
196. 6) {114} Начина по който се получава ефективният адрес **[От какво?]**
196. 7) {121} Намиране на точното място на данните в паметта,.
196. 8) {125} Това е нарушение на контрола по четност на ОП неизправен ЦП

197. 243. Каква е разликата между понятията „Режим на адресиране“, „Способ за адресиране“ и „Вид адресация“?

197. 1) {104} Разликата между „Режим на адресиране“, „Способ за адресиране“ и „Вид адресация“ са се в Режима на адресиране там е описано Режима по който е направена адресацията при Способа за Адресиране се разбира по какъв начин е осъществена адресацията и при „Вид адресация“ се разбира нейният вид. абсолютна или др.
197. 2) {105} Способ за адресиране – това е начина по който ще стане адресирането, чрез каква адресация? – това е вид на адресацията. Режим на адресиране – ни показва дали в дадения момент се изпълнява процеса адресация.
197. 3) {121} Способ за адресиране това е начин по които от битовете на

198. 244. Кога се приема, че даден ЦП притежава определен вид адресиране?

198. 1) {111} Зависи от кода на операцията
198. 2) {113} Видът на адресирането се определя от КОП
198. 3) {114} Приема се, че даден ЦП притежава определен вид адресиране
198. 4) {121} Зависи от кода на операцията.
198. 5) {122} Видът на адресирането се определя от КОП.
198. 6) {122} Когато самия ЦП е създаден за определен вид адресиране
198. 7) {124} Видът на адресирането се определя от КОП.
198. 8) {124} Когато ЦП се реализира чрез дадени от вид адресация
198. 9) {125} Когато ЦП работи само с определен вид адреси.
198. 10) {126} ЦП притежава определен вид адресиране, когато видът адресиране се обслужва от него Основно адресирането се определя от 2 независими признака – броя операнди, Адресирането бива еднокомпонентно и многокомпонентно

199. 245. Какво е предназначението (поне 3) на различните видове адресация?

199. 1) {105} Абсолютна – за достъп до клетки с фиксиран адрес (периферия)
Относителна с индексиране – за достъп до последователни клетки от паметта
Относителна по база с индексиране– за достъп до вектор
199. 2) {105} Различните видове адресация води до разграничаване на различните типове информация. По бърз достъп до информацията и нейното по добро структуриране в ОП.
199. 3) {105} Абсолютна адресация
199. 4) {106} – по-малко заемащо пространство
199. 5) – преобразуване на бази
– индексиране
199. 6) {113} Странично адресиране – резултата се описва на страници с равни размери, абсолютно адресите са зададени предварително, косвено – адресите се задават по време на изпълнението
199. 7) {113} а) къса абсолютна адресация – много адреси започват с 0
б) пълно абсолютно адресиране –
в) косвено абсолютно адресиране – динамична промяна на адреса
199. 8) {113} 1) да се разреши достъп на МИ до клетка на ОП
2) да се посочи пълен адрес от ОП с минимална загуба на битове съдържащи съкращения
3) за да се манипулират данни от стек или едномерен масив
199. 9) {124} – да бъде използвана за различни цели при нужда
199. 10) {125} Да се разреши на МИ да осъществи достъп до кретка от ОП, чиито адрес се изчислява в момента на изпълнение на щощомага. 2)Да се манипулира с адреси във форма, която е най-подходяща за често използвани структури от данни., 3) Да се посочи пълен адрес от ОП с минимално количество битове

200. 246. Какво представляват адресните регистри на ЦП?

200. 1) {103} идентифицират къде е записан обекта
200. 2) {106} Адресните регистри на ЦП помагат при адресиране на данните в ОП.
200. 3) {106} Посочват на кой адрес от паметта се записва информацията
200. 4) {111} Служат за запомняне (съхранение) на данните
200. 5) {121} Адресните регистри на ЦП се използват при изпълнение на МП за определяне на мястото на в ЦП по време на изпълнение.
200. 6) {121} Адресните регистри не посочват къде точно в паметта се намира дадена стойност
200. 7) {122} Регистри, които насочват централния процесор в кои адреси да запише информацията.
200. 8) {124} АР ни показват „кубчетата“ памет с тяхната позиция и съдържание.
200. 9) {126} Регистри, които имат свое уникално наименование.

201. 247. Как се формират адресни регистри, когато размерът на общодостъпните регистри на ЦП не е достатъчен за записване на пълен адрес от ОП?

201. 1) {105} Адресните регистри се формират като **адресите се записват в две клетки** по два начина
– Big Ending – **старшите** (левите) **са малки**
– Little Ending – **старшите** (левите) **са големи**
201. 2) {105} Прехвърлят се на старшия левия бит.
201. 3) {105} с допълнителни регистри
201. 4) {106} Формират се в **клетки, които могат да са различни по големина**
201. 5) {114} Предният регистър се прилепя за следващият.
201. 6) {124} Чрез служебни регистри.
201. 7) {125} Адресът се записва в 2клетки от регистъра на ЦП или „косвено“ се записва мястото където е записан целия адрес

202. 248. Какво представлява понятието „ефективен (изпълнителен) адрес“?

202. 1) {105} Адреса от който извличаме информацията
202. 2) {105} Това е адресът, от който ЦП търси готовият обработен на МЕ код.
202. 3) {106} Адрес в ОП, където се съхранява **МИ – резултат**
202. 4) {111} Точен адрес в ОП.
202. 5) {125} Това е адресът, който се използва за извикване на дадена променлива и съответната стойност.

203. 249. Какво представляват преките видове адресиране?

203. 1) {113} Преки видове адресиране – при търсене на даден адресне се чете целият му адрес а само няколко от неговите знаци, най-често използвани адреси са преки адреси.
203. 2) {121} Преки видове адресиране **в адреса** в паметта се зарисва информация, а не адрес, на който да се намери тя.
203. 3) {121} Преките видове адресиране са адреси, с които
203. 4) {123} преките видове адресиране могат да запазват и пренасят информация.
203. 5) {124} Преки видове адресиране
Повечето адреси на МИ започват с 0. Затова си записват без началните 0, и в процеса на извършване се допълват в зависимост от МИ.
203. 6) {124} Ефективният адрес определя положението
203. 7) {125} При тях адресирането се определя по местоположението

204. 250. Какво представляват косвените видове адресиране?

204. 1) {104} **Адресът е регистър** на ЦП, съдържа адрес към ОП, който се променя
204. 2) {105} При косвените видове адресиране изпълнителният адрес сочи адрес от ОП, където се намират операндите.
204. 3) {105} Неявно адресиране, най-често база – отместване и индекс
204. 4) {111} Адресация, при която адресът указва място в ОП, където е записана стойността.
204. 5) {113} Адресно поле задама мястото на клетката някаде в паметта
204. 6) {121} **На него** се записва адрес където да бъде намерена информацията
204. 7) {122} Косвени видове адресиране:
A → aB A → B
те не достъпват директно адреса на дадения процес а преминават през операнд който ги пренасочва към Адреса
Данни → операнд → Адрес
204. 8) {122} Видове които не могат да се променят . Имат постоянна стойност.
204. 9) {124} Ефещтивния адрес определя чрез посредничество чрез ОП или регистър къде да ъдат записани данните

205. 252. Възможно ли е един ЦП да предоставя само косвени видове адресиране? А полезно ли е?

205. 1) {123} Не е възможно
205. 2) {124} Генерирания ЕА се определя от един компонент. **При него адресирането** се задава **в него** или в апаратната част.

206. 253. Какво е характерно за еднокомпонентните видове адресиране?

206. 1) {105} Еднокомпонентните Видове адресиране са характерни, че при тях изпълненият адрес се задава по единствена стойност.
206. 2) {113} Адресирането може да започне от средата на паметта
206. 3) {113} Ефективният адрес се определя от от адресното поле и от регистър.
206. 4) {121} състоят се или от АП или от 1 регистър при тях изпълнителният адрес се задава по индексиране.
206. 5) {121} Може да се адресира само 1 компонент.
206. 6) {122} При еднокомпонентните видове адресиране е характерно, че на един адресотговаря само една програма.
206. 7) {123} Заемат по-малко място в паметта.
206. 8) {124} Характерно е, че всички данни се съхраняват на едно място
206. 9) {126} При него адресирането се задава или в апаратната част или в регистъра на ЦП
206. 10) {126} Генерираният ЕАсе определя от повече от 1 компонент. При него адресирането се задава или в апаратната част.

207. 254. Какво е характерно за многокомпонентните видове адресиране?

207. 1) {103} Състоят се от адресни полета и регистри. За получаване на изпълним адрес се комбинират указаните стойности.
207. 2) {103} Шанса за грешка и по голям, но едно адресиране върши повече работа.
207. 3) {104} АП + регистри
207. 4) {105} регистрова, абсолютна, непосредствен **оперант**, косвена регистрова, автоувеличение и автонамаление
207. 5) {113} Различните МЕ могат да имат операции, които не присъстват в останалите. Може да се различават и по начина на изпълнение на някои операции.
207. 6) {113} имат краен брой елементи. извличането на компоненти става чрез изместване равно на големината на даден тип
207. 7) {124} Те имат няколко компонента които помагат за по-точното намиране на определен адрес, за сметка на бързината.
207. 8) {125} Прилича на косвеното регистрово адресиране (КРА) Видовете адресиране биват – по база, по индекси

208. 255. Посочете известните ви еднокомпонентни видове адресиране (поне 5).

208. 1) {105} Пряко адресиране
Адресиране на база
Косвено адресиране

209. 256. Какво представлява регистровата адресация?

209. 1) {103} позволява достъп до ИП при елиминиране на 1-во адресно поле.
209. 2) {107} Съхранява изпълнението на адресните данни в ОП.

210. 257. Кога не може да се използва регистрова адресация? Защо?

210. 1) {105} Регистровата адресация не е приложима за преместване на данни в оперативната памет
210. 2) {110} При осъществяване на достъп към ОП, защото регистровите адресации работят само с регистри

210. 3) {111} Регистровата адресация работи само с регистри и затова не може при осъществяване на достъпа към ОП.
210. 4) {111} Когато искаме да достъпъм адрес от оперативната памет
210. 5) {114, 115} При осъществяване на достъп до ОП, защото регистровата адресация работи само за регистри
210. 6) {117} Дава възможност за формиране на къси инструкции, например един байт. Този байт указва както операцията така и определен регистър.
210. 7) {121} Когато суе осъществява в ЦП акумолатор за адресни регистри, защото тогава в адресното поле на МИ няма номер на регистъра да се запише.
210. 8) {122} При **осъществяването на ОП**, защото регистровата адресация работи само с регистри

211. 258. Каква е ползата от регистровата адресация?

211. 1) {104} Ползата от регистровата адресация е по-бързия достъп до информацията и при големите обеми от данни.
211. 2) {105} Премахва се първия операнд и се намалява размера на АП

212. 259. Кога се появява регистровият способ за адресиране?

212. 1) {104} Когато даден файл трябва да бъде преместен.

213. 261. Какви стратегии се следват при несъответствие между размерите на акумулатор и дума?

213. 1) {104} Думата може да бъде записана в последователност от клетки с поредни адреси

214. 262. Какво представлява абсолютната адресация?

214. 1) {103} Да се отиде директно безпрецедентно към зададена част от програмата независимо от обстоятелствата.
214. 2) {104} Когато **потребителят** се обръща към определен елемент от паметта с точен и пълен адрес
214. 3) {105} **Видовете биват** къса и странична абсолютна адресация. Късата се използва за съкращение на АП.
214. 4) {105} Абсолютната адресация; Прехвърлянето на информация от
214. 5) {106} Абсолютната адресация показва точно кои полета от паметта са нужни за изпълнение на задачата Те са едни и същи и никога не се променят. Така се постига по-бърз достъп до данните.
214. 6) {111} Използва се пълния адрес на следващата МИ
214. 7) {111} сектор + изместване
214. 8) {113} Адресно поле задава мястото на клетка в ОП. Това е най-естествената адресация
214. 9) {122} Това е когато клетката помни точния си адрес независимо от мястото където ще бъде преместена формулата, в която участва
214. 10) {125} Та има за цел да осигури достъп до адресното поле.

215. 263А. Какви са предимствата на абсолютната адресация?

215. 1) {104} Предимствата на абсолютната адресация са че благодарение на нея обработката на данни се извършва много по бързо
215. 2) {105} По-бърза но губим от точността
215. 3) {105} Абсолютната адресация е бърза и точна, но изисква повече място в ОП и заменя адреса.
215. 4) {106} Предимствата са: ясно се разчита; точна; и всичко се адресира.

216. 263В. Какви са недостатъците на абсолютната адресация?

216. 1) {118} Инструкцията е по-дълга
216. 2) {118} Включване на пълния адрес като съставна част от командата

217. 264. Избройте известните ви видове абсолютно адресиране.

217. 1) {123} Абсолютна адресация
– непосредствен операнд
– базова адресация (адресация по база)
– странична адресация

217. 2) {125} Пълна К.С. косвена

218. 267. Каква е целта на късата абсолютна адресация?

218. 1) {105} Целта на абсолютната адресация е да зададе директно адреса от оперативната памет

218. 2) {105} **Късата** абсолютна **адресация има за цел да посочи точно определено адресно поле**, докато косвената адресация посочва поле, в което се съдържа друг адрес.

218. 3) {105} Динамично модифициране на адрес.

218. 4) {106} Целта на късата абсолютна адресация е да се създаде възможно най-кратък път до данните в клетката като по този начин се ускорява процеса на работа Адресират се абсолютни данни

218. 5) {111} **скъсява** адресна та **проверка**

218. 6) {111} При ККА **[?!?]** се намалява броят на разредите за командата, но за сметка на това има пряк достъп само до ограничена част от адресното пространство

218. 7) {121} Целта на късата абсолютна адресация е спестяване на време при процеса на работа и намаляването на информацията за адресирането.

218. 8) {124} По бърз достъп до данните

218. 9) {125} Целта на абсолютната адресация по бързият достъп до данните съхранени в адреса.

219. 268. Какво е основанието за наличие на къса абсолютна адресация?

219. 1) {104} По-бързо откриване на търсения адрес.

219. 2) {104} Бърз достъп

219. 3) {104} за по-лесно намиране и боравене

219. 4) {106} Основанието е че всяка клетка има свой собствен номер.

220. 269A. До какво води наличието на къса абсолютна адресация?

220. 1) {102} Наличието на къса абсолютна адресация води до грешки в изпълнението на процеси.

220. 2) {107} При ККА се намалява броя на разряда за команда но за сметка на това има пряк достъп само до ограничена част на адресното пространство

220. 3) {107} Наличието на къса абсолютна адресация става привилегирована област на ОП
При ККА се намалява броят на разредите за командата, но за сметка на това има пряк достъп само до ограничена част от адресното пространство

220. 4) {109} Води до препълване

220. 5) {114} По бърз достъп до данните. Клетките се четат директно. Няма нужда да се адресират.

220. 6) {115} До трудност до достигане надресите и изпълнение на програмите, защото при къса абс. адресация много адреси започват с 0

220. 7) {115} По-къси програми и по-бързо изпълнение, защото АП са по-къси.

221. 274A. Посочете предимства на косвената абсолютна адресация.

221. 1) {104} Предимства на косвената абсолютна адресация е че информацията с предава по-бързо

221. 2) {106} Включване на пълния адрес, като съставна част от командата. Използват се много разряди за команди на ЕИМ. Разполагат с голямо адресно пространство. При косвената абсолютна адресация се намалява броят на

222. 274B. Посочете недостатъци на косвената абсолютна адресация.

222. 1) {106} При косвената абсолютна адресация има възможност за обхождане на вектори т. е. **изпълняват се две МИ едновременно** и има възможност за некоректен отговор (резултат)

223. 275. Каква е целта на адресация непосредствен операнд?

223. 1) {114} в адресното поле се записват данни вместо адрес
223. 2) {115} Да се ограничерче броят на Различните.

224. 276. Каква е същността на адресация непосредствен операнд?

224. 1) {119} Да облекчи задаването на констранти.
224. 2) {121} Да се облекчи задаването на константи
224. 3) {123} ако мястото в паметта не достигне се взимат се взики непосредствено до него адреси.
224. 4) {124} Адресацията с непосредствен операнд позволява по-лесно да извършваме някои от операциите като задаваме стойността непосредствено след командата
224. 5) {125} Да се облекчи задаването на константи.

225. 277. Абсолютно необходимо ли е ЦП да предоставя адресация непосредствен операнд? Защо?

225. 1) {105} Да защото ни трябва адреса да извлечем стойността зад него.
225. 2) {110} Да
225. 3) {123} Да необходимо е за да може да се следи последователноста на операндите.
225. 4) {124} Да необходимо е.

226. 277A. Абсолютно необходимо ли е всеки ЦП да предоставя адресация непосредствен операнд?

226. 1) {103} Да ЦП трябва да предоставя адресация непосредствен операнд
226. 2) {110, 110} Да

227. 277B. Защо не е абсолютно необходимо един ЦП да предоставя адресация непосредствен операнд?

227. 1) {121} адресация непосредствен операнд се използва за константи е възможно процесора да не „вижда“ такъв тип адресация
227. 2) {122} Защото този вид адресация е бавна.

228. 278. Винаги ли може да се използва адресация непосредствен операнд? Ако да – защо, ако не – кога не може да се използва такава адресация и защо?

228. 1) {115, 121} да
228. 2) {121} Да, винаги може, защото е нужна за използването и имплементирането на стек.
228. 3) {124, 126} Да
228. 4) {125} Тази операция облекчава използването на константи.на ги може да се използва, иначе програмирането ще е затруднено.

229. 280. Какво представлява косвената регистрова адресация?

229. 1) {103} При косвените адресации това **[Кое е това?]** с посредничеството на ОП или регистър. В командата се съдържа стойност на косвен адрес. Т. е. адрес на клетка от паметта, в който се намира окончателният изпълнителен адрес.
229. 2) {103} **Адресацията е в регистъра** на ЦП
229. 3) {111} **Адресът е регистър** в ЦП съдържащ адрес към ОП, който се променя
229. 4) {114} регистрена адресация с още една степен на косвеност
229. 5) {125, 125} Адресът е регистър в ЦП съдържащ адрес към ОП който се променя.
229. 6) {125} Адресът на регистър в ЦП, съдържащ адрес към ОП, който се променя.

230. 281. За какво най-често се използва косвена регистрова адресация?

230. 1) {114} При списък за обхождане на масив **[Така формулиран отговорът си е напълно грешен!]**
230. 2) {114} регистро, който съдържа адрес от ОП

230. 3) {126} Косвената регистрова адресацияа е вид еднокомпонентна адресация. Най-често се използва за списъци и едномерни масиви.

231. 284. Какви проблеми създава използването на косвена регистрова адресация при обхождане на вектор?

231. 1) {111} МИ се удължават и програмата се усложнява.
231. 2) {122} При косвената регистрова адресация обхождането на вектор е трудно и дълго и има възможност от забравяне до къде е стигнало обхождането.

232. 285. Какво представлява адресация „авто-увеличение“?

232. 1) {102} Авто увеличението е процес на добавяне на нови клетки към записа при запълване на паметта. Новите клетки се адресират автоматично на следващото свободно поле. Това действие се предприема от системата като но е с избрано предварително в останалите ситуации с-мата подава сигнал за OVERFLOW
232. 2) {110} адресацията авто-увеличение посочва следващата клетка от паметта
232. 3) {111} Степен на косвеност **спрямо автоувеличение**
232. 4) {111} адресация авто увеличения обема адреса е на адресно поле по който битове поле ще бъди определен адрес
232. 5) {117} Запазва старото и прибавя още едно
232. 6) {121} При обхождане по намаляване на адресите за обработка на данни. Данните се намаля.
232. 7) {121} Процесът на адресация се извършва автоматично, като всеки един адрес приема стойност базирана на предходния
232. 8) {122} Адресацията авто-увеличение представлява автоматично да се увеличат броя на думите при адресация на междинните данни.
232. 9) {123} Когато може да изпълни МИ (Машинна Инструкция) изискваща съответната адресация
232. 10) {124} Авто-увеличение – използваната стойност се променя като се увеличава стойността на данните от адреса.
232. 11) {125} Целта е да се улесни обхождането на таблици с указатели към данни Те са косвени видове адресиране.
232. 12) {125} Модификация на косвена регистрова адресация. Изролзва се за обхождане на последователни елементи.
232. 13) {125} Когато нужните ни ресурси са записани последователно в паметта, тогава когато знаем адреса на първия може с увеличаване на този адресда достигнем следващите ресурси

233. 290. Какво представлява адресация „авто-намаление“?

233. 1) {109} Точно обратното на авто-увеличението. След използването на стойността на регистъра, същата не се променя с добавяне на размера на данните (в брой клетки)
233. 2) {111, 111} Тя е точно обратната на автоувеличението.
233. 3) {111} Посочване на мястото на даден обект в паметта.
233. 4) {111} Облекчава пренасянето на блок от ОП.
233. 5) {113} Обратна на авто увеличението.
233. 6) {114} Обратното на автоувеличение. Модификация на косвена регистрова адресация.
233. 7) {114} Точна обратна на „авто-увебичаване“
233. 8) {115} Адресацията „авто-намаление“ не може да се изпълни без да бъде изпълнена тази за „авто-увеличение“.
233. 9) {121} Подобно на авто увеличаване на в този случай намалява взетоте място при обработка на данни от различен или еднакъв тип. Примерно от десетичен в двоичен код.
233. 10) {122} След използване на стойността на регистъра, същата се променя с намаляне на размера на данните (в брой клетки)
233. 11) {124} Адресация при която всеки следващ адрес се намалява с 1 бит
233. 12) {125} Адресацията „авто–намаление“

233. 13) {125} Тя е точно обратната на автоувеличението.
233. 14) {125} Това е вариант на косвена региструема адресация при която се получава ЕА и стойността на регистъра се намалява с размера на данните
233. 15) {126} Решава проблема при обхождане на някои от структорите на данни (списъци и едномерни масиви
233. 16) {126} Авто-намаление представлява противоположното на авто-увеличение

234. 295. Какви проблеми могат да бъдат решени при едновременно наличие на „авто-увеличение“ и „авто-намаление“?

234. 1) {111} Изгражда се **стек от ОП**.
234. 2) {111} Едновременното използване на две МИ и на стек.
234. 3) {122} Те са синоними, след използване стойността ва регистъра се променя с добавяне размера на данните – „авто-увеличение“
234. 4) {124} Отмесдтване наляво и надясно

235. 296. Ка кво решение предлагат архитектите на ЦП без адресации „авто-увеличение“ и „авто-намаление“? Защо?

235. 1) {122} Повече ядра в зависимост от натоварването на ЦП за авто-увеличение и авто-намаление.

236. 297. Какво представляват косвените варианти на адресации „авто-увеличение“ и „авто-намаление“?

236. 1) {107} Те са косдвено косвени видове на автонамалението и автоумеличението.
236. 2) {107} Те имат още една степен на косвеност спрямо авто-увеличението и авто-намалението
236. 3) {110} Имат още една степен на косвенос към нормалните авто-увеличение и авто-намаление
236. 4) {112} Те имат още една степен на косвеност спрямо амто-увеличение и авто-намаление.
236. 5) {114} Косвените варианти на адресация „авто-увеличение“ и „авто-намаление“ помагат (имат за цел)да се улесни обхождането на таблица с указатели към данни

237. 301. По какво се различават прякото и косвеното „авто-увеличение“?

237. 1) {110} Прякото увеличение се извършва от програмиста, а косвеното от вътрешните нужди на програмата
237. 2) {115} Косвената е още една степен на косвеност на пряката. При пряката изчислев адрес сочи директно данни в ОП, а при косвена сочи в ОП адреса на данни.
237. 3) {115} Косвеното автоувеличение има още една степен на косвеност.
237. 4) {119} Прякото увеличение има още една степен на косвеност спрямо нормалното.
237. 5) {112} Косвеното авто-увеличение добавя още една степен на косваност, което е полезно при обхождане на таблица от указатели
237. 6) {121} Прякото авто-увеличение се извършва директно, а косвеното се извършва от външна променлива
237. 7) {122} Различават се по това че при прякото „авто-увеличение“ се работи изарежда цялата информация. При косвеното се зарежда част с цел бързодействие.
237. 8) {122} Косвеното авто-увеличение има още една степен на косвеност спрямо обикновеното.
237. 9) {122} косвеното има една повече степен за косвеност

238. 303. По какво се различават прякото и косвеното „авто-намаление“?

238. 1) {115} Различават се по това, че прякото авто-намаление премахва операнда записан в регистъра на ОП
238. 2) {124} Различават се по това че при прякото на адреса, записан в регистъра има операнд докато при косвеното има друг адрес.
238. 3) {124} Различават се по това, че при прякото „авто-намаление“ на адреса, записан в регистъра има операнд, докато при косвеното има друг адрес

238. 4) {126} Различават се по това, че при прякото на адреса, записан в регистъра има операнд докато при косвеното има друг адрес
238. 5) {126} Косвеното авто-намаление има още една степен на косвеност спрямо обикновенното
238. 6) {126} Обратното на „авто-увеличение“

239. 304. Съществуват ли еквивалентни видове еднокомпонентни адресации? Ако не – защо, ако да – посочете ги и обяснете защо са еквивалентни.

239. 1) {113} Бърз достъп до данни

240. 305. Посочете известните ви многокомпонентни видове адресиране (поне 4).

240. 1) {103} Абсолютна; непосредствен оперант, авто-увеличение; авто-намаляване, косвено авто-увеличение; косвено авто-намаляване
240. 2) {114} Абсолютна адресация, адресиране чрез номериране на клетките.

241. 306. Каква е целта на стрàничната адресация?

241. 1) {110} Бързодействие.
241. 2) {110} Целта е по-бързо достигане до дадено АП.
241. 3) {112} Целта е динамична промяна на адреса
241. 4) {121} Страничната адресация е с цел улесняване на програмиста. **Тя може да бъде и като коментар**
241. 5) {121} Отделяне от останалите адреси
241. 6) {121} Да спомага за правилното извършване на адресацията.
241. 7) {122} Динамично модифициране на адрес
241. 8) {122} Целта на страничната адресация е да предпази
241. 9) {124} Чрез нея можем да намерим определена информация в случай, че другата адресация е непълна или неясна
241. 10) {124} Страничната адресация позволява лесно посочване и адресиране на страници
241. 11) {125} Това са регистрите с общо предназначение:
– акумулатори
– инзексен, указател на стека

242. 307. Какъв е основният принцип на стрàничната адресация?

242. 1) {107} Операция която инвертира всички битове на дадена дума
242. 2) {110} Адресирането се осъществява като се задава адресът на страницата и отместването в нея.
242. 3) {111} Адреса се записва в първия бит на клетката
242. 4) {119} Основният принцип на страничната адресация е динамично модифициране на адреса
242. 5) {121} Съкращаване на адресно поле
242. 6) {123} Основният принцип е да подпомогне и в случай че има нужда да допълне
242. 7) {124} съкращаване на основните полета.
242. 8) {125} съкращаване на адресни полета. .

243. 308. Какви варианти на стрàнична адресация съществуват?

243. 1) {104} Пълен и непълен път на странична адресация.
243. 2) {110} Пряка, Коствена адресация
243. 3) {121} Къса, пълна
243. 4) {123} нулева, единична
243. 5) {123} Десетичните числа се представят с четири цифри – двоично кодирани десетични числа
243. 6) {124} Векторен и растерен формат

244. 313A. Какви са предимствата на стрàничната адресация в текущата страница?

244. 1) {105} Предимствата на страничната адресация в текущата страница са следствие на директорията където се извършват операции
244. 2) {105} Предимствата на страничната адресация в текущата страница са: указват на директорията където се извършват операции
244. 3) {105} Улеснява работата на оператора.
244. 4) {111, 111} Адресното поле съдържа само отместването
244. 5) {115} ① В страница нула (0);
② В текуща страница;
③ Чрез страничен регистър.
244. 6) {118} Предимството е, че адресното поле съдържа само отместването, недостатък е, че програмният брояч участва в образуването на ефективния адрес
244. 7) {120} Чрез нея можем по-бързо да стигнем до информацията, която търсим при по-голям обем от данни
244. 8) {121} съкращаване на **адресната лента**.
244. 9) {121} Предимствата са , че може да се съберат повече адреси настрани.
244. 10) {122} разширява се адресното пространство
244. 11) {125} Предимството е, че адресното поле съдържат само отместването.
244. 12) {125} Съкращаване на адресното поле
244. 13) {126} Достъп до ефективен масив „списък“ и други структури от данни по време на изпълнение
244. 14) {126} Предимствата са, че има по-бърз достъп

245. 313B. Какви са недостатъците на стрàничната адресация в текущата страница?

245. 1) {105} Страничната адресация в текущата страница създава проблеми при търсене
245. 2) {113} програмният брояч участва в образуването на ефективен адрес
245. 3) {121} Програмата се пише трудно и бавно, промяната е сложна и грешките се укриват
245. 4) {121} Програмният брояч участва в образуването на ефективния адрес
245. 5) {122} Тя е най-дълга.
245. 6) {122} Малко пространство и има възможност за обърквање на страницата
245. 7) {124} Предимствата е че адресното поле съдържа не само отместването. Недостатък е че програмния брояч участва в образуването на ефективния адрес.
245. 8) {126} Недостатък е това че програмния брояч участва в образуването на ефективния **процес**

246. 314. Каква е целта на адресацията с индексиране?

246. 1) {102} Да се ускори работата на процесора.
246. 2) {105} адресните полета получават определен индекс
246. 3) {107} динамично
246. 4) {109} Възможност за запис в повече от една клетка
246. 5) {111} Базов адрес в АП се модифицира с добавяне на отместване в регистъра на централния процесор
246. 6) {111} целта на адресация с индексиране е по-лесен достъп до данните
246. 7) {112} по-бърз достъп
246. 8) {113} индексира адреса в ОП
246. 9) {114} По-бързо прочитане на информацията спрямо определения индекс
246. 10) {114} Целта е по-бърз и лесен достъп до данните
246. 11) {115} Да адресираме с индекс
246. 12) {119} За да се избегне забавянето при адресация и да се ускори **[Не се чете!]**
246. 13) {121} Целта на адресация с индексиране е да може лесно да се намери „дума“ записъна в паметта. Думата се записва последователно в паметта и за да я намерим трябва само да намерим първата клетка в паметта!

246. 14) {121} Целта на адресация с индексиране е подобряване на подредбата на различни стойности или най-главната цел която се извършва е структуриране по ред, дума, стойност тоест постига се някакъв ред
246. 15) {123} За да може по лесно и бързо нужния **файл или функция**. Това намалява и разходи на ресурси.
246. 16) {124} Целта на адресация с индексиране е за да внесе памет и за да добърза скоростта на работа
246. 17) {126} използва се в ОП за обръщение към блок данни

247. 315. Какво представлява адресацията с индексиране?

247. 1) {102} Адресацията с индексиране представлява изпращане на данни с индекс за изпълнение.
247. 2) {102} задават се индекси на операциите. За по-бедно и бързо извикване/обработване.
247. 3) {105} Адресация с индексиране – при тази адресация в адресното поле на машинната инструкция се поставя индекс, който посочва адреса (същинския адрес) на необходимите данни
247. 4) {115, 124} Динамично модифициране на адрес
247. 5) {122} Индексни регистри
247. 6) {124} Адресите в ОП се номерират за да бъдат намирани по-лесно и бързо и за да се различават
247. 7) {124} Да разделя с индекс даденото условие
247. 8) {124} Вариант на косвена регистрова адресация. При нея след използването на стойността на регистъра, тя се променя с добавяне на размера на данните (в брой клетки) Също така целта е – Динамичното му модифициране на адрес. Това е базов адрес в адресното поле, който се модифицира с добавяне на регистър на ЦП.
247. 9) {125} Динамичното модифициране на адрес
247. 10) {125} Индексни регистри
247. 11) {125} Адресацията с индексиране представлява поредно номериране на елементите.

248. 317. Как е прието да се наричат регистрите, използвани при адресация с индексиране?

248. 1) {106} Регистрите използвани **при адресиране по база се наричат индексни** регистри
248. 2) {110} **Базови** регистри.
248. 3) {114} Индексирани регистри
248. 4) {122} Косвени регистри.
248. 5) {124} Адресни регистри
248. 6) {124} индекси
248. 7) {124} Stack (стек)

249. 320. Каква е целта на адресирането по база?

249. 1) {104} Целта на адресирането по база е че достъпът е по бърз
249. 2) {104} За да може дадена функция да бъде изпълнена от дадена база, без самата функция да се намира в базата.
249. 3) {113} Адреса минава през цялата база и на изхода извежда резултат
249. 4) {114} Целта на адресирането по база е съкращаване
249. 5) {115} Целта на адресирането по база е да се осигури достъп до поле от запис
249. 6) {124} По-бързотта, защото се елиминира възможността за грешки.
249. 7) {124} Базов регистър посочва началото или средата на достъпната област от ОП, които са съсредоточени в малка област от адреси.

250. 321. Какво представлява адресирането по база?

250. 1) {103, 103, 106, 114, 123, 125, 126} Базов регистър посочва началото или средата на достъпната област от ОП

250. 2) {105} Адресирането по база представлява това че обръщенията към ОП памет са съсредоточени в по-малка област от адреси. Показва началото или средата на достъпната област от ОП.
250. 3) {105} **Адресирането** по база **сочи към определена част от паметта**. Адреса може да сочи към началото или средата на тази част.
250. 4) {105} Началото на програмата се назначава базов адрес и всички останали адреси се изчисляват по отместване т. е. прибавяне на относителния адрес към базовия. Позволява програмата да се зарежда на произволно място в паметта.
250. 5) {111, 122, 122} Базов регистър посочва началото или средата на достъпната област от оперативната памет. (ОП) **[Е и?]**
250. 6) {123} Базов регистър и отместването определят
250. 7) {124} Адресирането по база представлява база от данни всеки сочещ на някъде и всеки може да извлича независимо позицията му без забавене.
250. 8) {125} Адресиране с пълен базов регистър. При него има „икономичност“ Стойностите му не се променят.

251. 322. На какво се основава адресирането по база?

251. 1) {104} Адресирането по база чрез запомняне на адреса на клетката която е в началото. Има два начина на запомняне:
1 – базата се записва в АП, а отместването в регистър на ЦП.
2 – базата се записва в регистър на ЦП, а отместването **в АП на клетка**.
251. 2) {124} Адресирането става посредством кода на операцията.
251. 3) {126} Базов регистър посочва началото и средата на достъпната област в ОП.
251. 4) {126} Основава се на обръщение към АП, който е съсредоточено в малка област на адреси

252. 324. Как е прието да се наричат регистрите, използвани при адресиране по база?

252. 1) {102} Регистрите при адресиране по база наричаме листове (Sheets) като за отделните листове задаваме различни номера и ключове.
252. 2) {102} Адресация (адресна база)
252. 3) {113} Индекси.
252. 4) {113, 121} Адресни регистри
252. 5) {124} Позиционни, като статично позиционните могат да бъдат използвани и при адресиране по база , и при относително адресиране.

253. 325. Кога е най-изгодно да се използва адресиране по база?

253. 1) {104} Адресирането по база е най-изгодно когато клетките съдържат подобни по обем и съдържание данни които могат да се групират към общи бази.

254. 326A. По какво се различават адресирането с индексиране и адресирането по база?

254. 1) {105} Адресиране – процеса на насочване към конкретния адрес
Индексиране – самият адрес в ОП.
254. 2) {105} Адресното индексиране
Адресирането па база

255. 326B. По какво си приличат адресирането с индексиране и адресирането по база?

255. 1) {104} По началното извикване
255. 2) {106} По това, че се преобразуват и адресират в дадена база
255. 3) {113} по нищо
255. 4) {114} Съставени са от две компонента на АП и регистър. на адресното поле.
255. 5) {114} Приличат си по това че и двата начина са в услуга на потребителя.
255. 6) {119} И двата варианта са полезни – (адресирането) позволяват да се променя адресът промяна на отместването. Прединдексна косвеност – комбинация от базово адресиране и

отместване сочи към клетка с адрес Слединдексна косвеност – базовият е косвен адрес, отместването от индексния регистър дава ЕА на клетката с която ще работи.

255. 7) {121} тези намини на адресиране използват различни начини на действие, но се намират заедно често, за постигане на желан резултат

256. 327. Какво представлява адресирането по база с индексиране?

256. 1) {102} Базов регистър посочва началото или средата на достъпна областът ОП.
256. 2) {111} Базов адрес в АП се модифицира с добавяне на отместване в регистър на ЦП.
256. 3) {114} Адресирането по база с индексиране представлява адресиране на една клетка до друга в регистъра спрямо индекс
256. 4) {114} Записване на адреса в индекс
256. 5) {114} използват се **базови индекси** индекса се съставя като **збор** от базата плюс индекса
256. 6) {114} Съкъсяване на адресно поле
256. 7) {118} Адресирането по база с индексиране има за цел да намери адреса на дадена клетка в паметта.
256. 8) {123} Всеки адрес се обозначава с определен за него индекс. Зостъпът до този адрес се осъществява, като се намери по индекса.
256. 9) {124} те определят викторът и индексирането
256. 10) {125} Адресиране по индекс.

257. 330. Как се наричат регистрите, използвани при адресиране по база с индексиране?

257. 1) {121} База от Данни.
257. 2) {122} **Вектери**

258. 331. Каква е целта на относителното адресиране?

258. 1) {104} Целта на относителното адресиране е относителен адрес и абсолютна стойност
258. 2) {105} Целта на относителното адресиране е клетките се предават групово, като се помни само адреса на началната клетка и на файл
258. 3) {105} Относителното адресиране е вид адресация, при което с референцията се към друг адрес, а не към конкретна клетка в паметта.
258. 4) {106} Целта на относителното адресиране е възможност да се прекрати програма в определен момент, да се изпълни друга програма (или ППГ) – подпрограма и после да се продължи изпълнението на временно прекратената. Освен това с относително адресиране по-лесно се обхождат масиви и има преимущества при стекова организация на данни.
258. 5) {107} Адресността на дадена машинна посочва броя на нейните адресни полета.
258. 6) {109} Целта на относителното адресиране е бързодействието
258. 7) {112} При относителната адресация, в адресната част на командата не се намира нито самия операнд, нито неговия изпълнителен адрес, а адресен указател къде се намира адреса на операнда.
258. 8) {114} Странична, с индексиране, по база, по база с индексиране, относителна.
258. 9) {121} При относителното адресиране в паметта на компютъра се записва адрес на който да бъде намерена информацията
258. 10) {122} Целта на относителното адресиране е бързото разпредел(яне на данните в адрес от Оперативната памет (ОП) и ОП ще знае какви данни са съхранени в адресите ѝ (отделните ѝ адреси)
258. 11) {124} по-лесно намиране на клетката и използването ѝ

259. 332. Какво представлява относителното адресиране?

259. 1) {111, 114} Относителното адресиране цели да определи адрес относно МИ. Прилагат се за управляващи МИ. 90% от преходите са наблизо.
259. 2) {112} Записване на символично име в полето за етикет или при дефиниране на данни в паметта и запазването на памет за общо ползване.
259. 3) {122} Да определи адрес относно МИ

259. 4) {122} Посочва точния адрес
259. 5) {123} Адресиране – адресът съдържа указател към клетка в паметта
259. 6) {124, 125} Относителното адресиране **цели** да определи адрес относно МИ **Прилага се за управляващи МИ** Тук 90% от преходите са наблизу.
259. 7) {125} Стекът се почиства от активиращите след възврат от всяка подпрограма, защото подпрограмите могат да се викат с различен брой параметри.
259. 8) {125} Относителното адресиране представлява образуване на ефективен адрес чрез отместване спрямо някаква база (спец. регистър)
259. 9) {125} При относителната адресация може да декларираме временни стойности, които впоследствие могат да бъдат променяни
259. 10) {125} Адресиране спрямо друг адрес.

260. 334. Допуска ли относителното адресиране някакви модификации? Ако не – защо, ако да – кога и какви?

260. 1) {104} Не се допуска относително адресиране защото се допускат грешки с мястото за адресиране
260. 2) {110} Не

261. 335. Какво представляват позиционно независимите МП?

261. 1) {105} Позиционно независимите МП са машинни програми, които могат да бъдат изпълнени от различни места в ОП
261. 2) {105} При тях записа на МП не е обвързано с конкретен адрес. Те могат да се зареждат в паметта на произволно място.

262. 336. Какви видове позиционна независимост на програмите съществува?

262. 1) {103} Съществуват два позиционна независимост
1) статична, където програмата може да бъде въведена на произволен адрес на ОП и
2) динамична по време на изпълнение.
262. 2) {106} Package, class

263. 337. Какво представлява статичната позиционна независимост на една МП?

263. 1) {102} Вместо номерата на адресите се използват техните имена
263. 2) {102} когато програмата може да бъде въведена (изведена) на произволен адрес, говорим за статична позиционна независимост
263. 3) {103} Когато програмата може да бъде въведена на произволен адрес от паметта, говорим за статично позиционна независимост.
263. 4) {103} Програмата може да бъде въведена на произволен адрес от ОП
263. 5) {104} изпълнението на МП не зависи от
263. 6) {105} ~~Регистир~~ → ROM → памет
263. 7) {106} Позиционнато независимост на една МП означава, че достъпът до нея не зависи от местоположението ѝ в ОП
263. 8) {115} Програмата може да бъде въведена на произволен адрес на ОП **[Няма програма, която да не може да бъде въведена на произволен адрес от ОП!]**

264. 338. Какво представлява динамичната позиционна независимост на една МП?

264. 1) {104} Когато по време на изпълнение прогр. може да бъде преместена, тогава имаме динам. позиц. независимост
264. 2) {105} По време на изпълнение програмата може да бъде преместена
264. 3) {106} Динамичната позиционна независимост на една МП се състои в това че МП може да бъде изпълнена не само на един конкрете ЦП, а тя (МП) може да бъде преместена на различни

265. 339. Как може да се създаде статична позиционно независима програма?

265. 1) {103} Основната характеристика на статичната позиционно-независима програма, е че в нея не се съдържа абсолютни адреси.
265. 2) {103} като не се използва абсолютна адресация
265. 3) {105} Чрез функцията „static“
265. 4) {105} Позиционно независимата програма дава възможност тя да бъде заредена на произволно място в ОП
265. 5) {105} Основната характеристика, че те не съдържат **основни** адреси.

266. 340. Кои видове адресиране могат да се използват и кога за да бъде създадената програма статично позиционно независима?

266. 1) {103} Абсолютно и по база, като се използва само за външни за програмата адреси
266. 2) {104} Използване на стек.
266. 3) {105} Статично позиционно независима програма е такава, която може да бъде заредена от произволен адрес на Оперативната памет, затова видовете адресиране, подходящи за тази програма са: регистрова адресация, индексна адресация
266. 4) {105} Може да се използва пряко адресиране Използваме пряко адресиране когато искаме да заделим точно избраната от нас част в паметта а от там да разположим статично позиционната независима програма
266. 5) {105} Адреси за достъп на данни от ОП, има адреси за ППГ

267. 341. Как се постига динамична позиционна независимост?

267. 1) {106} Позиционна независимост се постига чрез стекова организация на данните. (динамична)

268. 342. Какви са функциите на блока за преобразуване на адреси?

268. 1) {105} Блока за преобразуване на адресите се използва за работа на ЦП с области от паметта с по-големи адреси. Адресацията се увеличава например от 16 до 20 бита чрез таблица.

269. 344. Какъв е принципът на работа на блока за преобразуване на адресите?

269. 1) {105} БПА получава от ЦП логически адрес, които е разделен на две части – първата част е адрес на сегментен регистър, втората част е отместване. По този начин БПА адресира части от ОП, които ЦП директно не може да адресира. Сегментните регистри са 16 на брой и образуват карта на паметта.
269. 2) {105} Някой КС използват блок за преобразуване на БПА, увеличаване на ОП, над този на И.П. Разделяне на ОП.
269. 3) {106} Принципът е, че всеки адрес се взема и се преобразува. Това става, за да бъде улеснена работата.

270. 346. Как се класифицират машинните операции, реализирани чрез МИ?

270. 1) {104} по-степен на важност

271. 347. Какъв е форматът на МИ с 1 операнд?

271. 1) {104} Форматът е **реално число**
271. 2) {105} <Инструкци> <Адрес> <операнд>
271. 3) {105} Форматът на МИ в I операнд е в двоична бройна система
271. 4) {106} Като | инструкция | ОПЕРАНД I | ЗАПИС |
| четене | | |

272. 348. Какви действия извършва ЦП при изпълнение на МИ с 1 операнд?

272. 1) {102} Четене,запис. **[На какво?]**
272. 2) {105} ОП приемник.
272. 3) {107} Чете се израза, и аритметико-логическото устройство извършва действията.

272. 4) {112} Първо прочита МИ сред това МЕ, след което изпълнява МИ записва данните и продължава със следващата МИ.
272. 5) {123} Прочита МИ след което използвайки МЕ взема решение и изпълнява МИ с 1 операнд.

273. 350. Какви действия се извършва ЦП при изпълнение на МИ с 2 операнда?

273. 1) {103} Използване на аритметико-логическо устройство за пресмятането и запис на резултата
273. 2) {105} Първо изпълнява изчисленията с по-голям приоритет, докато другите ги стопира и след това, като приключи започва с тях
273. 3) {109} Първо прочита МЕ който трябва да използва и запазва място в ОП на което да работи, и изпълнява МИ.
273. 4) {114} Четене – транслация – изпълнение – запис
273. 5) {115} Прочита се код на операцията, операнд 11 операнд 2, обработка и връщане на резултат

274. 351. Как изглеждат МИ на акумулаторните процесори?

274. 1) {105} МИ (къде), (какво) най-често МИ (регистър)(памет) но при някои МИ може и (памет)⇒(памет)
274. 2) {106} В по-голям обем и размер на данните

275. 352. Как изглеждат МИ на процесорите с РОП?

275. 1) {104} МИ на процесорите с РОП имат универсални адреси, което дава възможност броят им да бъде по-голям в сравнение с уникалните адреси. Недостатъкът е, че универсалните адреси трябва да се посочват и да се разпределят за какво ще бъдат използвани.
275. 2) {115} По време на изпълнение на МИ от ОП се четат операндите и се прехвърлят в РАП АЛУ изчислява резултата на базата на заведените операнди и операцията.

276. 353. Какво представляват МИ тип памет–памет?

276. 1) {102} МИ представляват памет в която се съхраняват данни за изпълнението и начина на на изпълнението на даден Машинен Процес

277. 354A. За какво е необходим РУ на ЦП?

277. 1) {105} РУ е необходим на ЦП, **за да извършва своите действия**.
277. 2) {110} Регистъра на условието съхранява междинна информация
277. 3) {110} РУ е необходимо, за да следи дали са налице условия, поради които трябва да се прекъсне изпълнението на текущата машинна инструкция.

278. 354B. Какво представлява РУ на ЦП?

278. 1) {119} Регистър на условията в ЦП Съвкупност от бити, всеки регистрира наличие на определено условие, което е независимо от другите условия.
278. 2) {119} РУ – регистър на указатели

279. 355. Как може да бъде изработен РУ на ЦП?

279. 1) {105} РУ на ЦП може да се изработи апаратно или микропрограмно. Апаратно е решението при RISC процесорите а микропрограмно е при CISC.
279. 2) {105} Като **масив от данни или таблица**.
279. 3) {115} Като част от ЦП.

280. 356. Кой вариант на РУ се практикува при съвременните микропроцесори?

280. 1) {110} регистър с общо предназначение

281. 357. Опишете основните флагове в РУ (поне 3).

281. 1) {103} Броят названието и предназначението на разрядите за условията са различни PDP11, M68000, I80x86

281. 2) {109} Подготовка на процеса, изчакване, изпълнение, преповаке на процеса
281. 3) {115} честитност, разширение, полупренос, посока2) (1401417004 Атанас Трайков Авков)

282. 358. Опишете поне три допълнителни флага в РУ.

282. 1) {117} * нулев резултат (Zero, Zflag): $1 \leftrightarrow =0$, $0 \leftrightarrow \neq 0$
* знак отрицателен знак (Sign, Negative, SF
* пренос (Carry, CF) = препълване без знак

283. 359. Посочете функциите на флаг C (Carry/Borrow = Пренос/Заем) в РУ.

283. 1) {117} Флаг „C“ се активира ако има пренос при събиране на числа.
283. 2) {119} Функциите на флаг C (Carry/Borrow) в РУ са
– Регистрира дали има или няма пренос за разрядната решетка
– Наличието на пренос след операция на числа с знак не значи че има препълване, то се установява ако например при две числа с един и същи знак се получава друг знак за резултата.
283. 3) {119} Регистрира дали има или няма пренос от разрядната решетка.

284. 361. Необходимо ли е ЦП да предоставя операции за работа с РУ като цяло? Обосновете отговора си.

284. 1) {104} Да ANDCC логическо И с РУ (нулиране); ORCC – логическо Или с РУ(запис с единици) PUSHCC – запис на РУ стека. PULLCC четене РУ в стека
284. 2) {105} Да, нужно е, за да знае АЛУ какво трябва да върне като резултат
284. 3) {106} Да. ANPCC – логическо И с РУ (НУЛИРАНЕ); ORCC (логическо или) с РУ (запис с единици); PUSHCC – запис на РУ в стека PULLCC – четене на РУ в стека

285. 362. Как може да се определи кои МИ са по-необходими за писане на компютърни МП?

285. 1) {107} Чрез симулации на ЦП може да се определи кои операции **по-често се изпълняват** от него
285. 2) {107} Чрез сиизлация на ЦП може да се определи кои операции по-често се изпълняват от него

286. 364. Какво влияние оказва използваемостта на дадена група операции при конструиране на МЕ?

286. 1) {108} Тъй като при математическите съпроцесори адресното поле за другия операнд е свободно обикновено всички операции имат еднакъв КОП, я това поле доуточнява действието им.
286. 2) {110} Това е работа за един такт на процесора, т. е. една МИ (машинна инструкция)
286. 3) {112} от съществено значение при създаване на МЕ е правилното кодиране на операциите.
286. 4) {114} от съществено значение е правилното кодиране на операциите. За това се използват сведения за това кои операции се използват по-често при писането на програмите

287. 367. Коя е най-използваната група от операции в МЕ?

287. 1) {110, 110} Събиране и изваждане

288. 368. Какво представляват операциите с повишена точност?

288. 1) {118} Повишена прецизност на резултата удвоена точност реализира се с над 32 бита, отчита се флаг C
288. 2) {118} Операции при които се взима в предвид флага за пренос C (Carry Flag)

289. 375. Какво представляват математическите съпроцесори (копроцесори)?

289. 1) {112} Копроцесорът изчислява математическите инструкции (операции) на процесора.
289. 2) {114} Математическите съпроцесори се използват в решаването на сложни диференциални уравнения,

289. 3) {123} Помощни процесори, които обработват специфични математически изчисления

290. 376. Какви операции реализират математическите съпроцесори?

290. 1) {112} Математическите съпроцесори осъществяват операцията събиране.
290. 2) {118} Математическите съпроцесори могат да реализират паралелни операции.

291. 388А. Как обикновено се наричат управляващите МИ?

291. 1) {102} машинен код

292. 390. По какво управляващите инструкции съществено се различават от обработващите? Защо?

292. 1) {105} Управляващите инструкции задават какво трябва да се свърши, а обработващите извършват действията, следователно управляващите задават каква е следващата обработваща инструкция
292. 2) {106} Управляващите инструкции се различават от останалите по дейността, която изпълняват. Управляващите инструкции спомагат за посочване, насочване и контролиране на процесите докато обработващите обработват вече зададената от управляващото устройство информация

293. 391. Каква е разликата между МИ за безусловен преход и тази за безусловен преход с възможност за възврат?

293. 1) {105} При възможност за възврат след изпълнението на МИ може, да се върне на определено място, което е записано предварително в регистрите
293. 2) {106} Машинните инструкции за безусловен преход се изпълняват от устройството като от процеса на изпълнение командите се запаметяват без възможност за промяна Тези с възможност за възврат позволяват да се нанасят корекции като в процеса на изпълнение текущо се връща към дадена команда

294. 392. Как често се нарича МИ за безусловен преход с възможност за възврат? Защо?

294. 1) {105} Възвратна МИ

295. 397. Дайте дефиниция на понятието „подпрограма“.

295. 1) {104} Подпрограмата е част от програмата, която се извиква и изпълнява при нужда при дадени условия.
295. 2) {104} Подпрограма наричаме програма, която съдържа в себе си стойности от главната програма
295. 3) {105} Подпрограма – алгоритъм, който е написан веднъж, но може да бъде извикан многократно от различни места на главната програма. Разполага се на едно място в ОП.
295. 4) {105} Логически обособена част от програма с начало и край, входни и изходни параметри Синоними: процедура, функция.
295. 5) {110} Програма извикана от главната програма
295. 6) {115} Подпрограмата е съвкупност от МИ определена и съхранена в ОП
295. 7) {118} ППГ определят до голяма степен структурата на програмата при всеки език, без оглед на равнището му.
295. 8) {121} Подпрограмите най-често са функции и процедури. Те решават даден проблем, като най-характерното за тях е тяхната многократна използваемост в дадена програма.
295. 9) {123} Под-програма е алгоритъм за улеснение на дадено действие от програмата чрез рационализиране и следена за специфично действие което може да доведе до грешен резултат.
295. 10) {124} Последователност от МИ, която е определена и се съхранява на едно място в ОП.
295. 11) {124} Подпрограмата е действия (функция) която е независима от главната но може да бъде извикана в нея многократно с различни входни данни или без такива
295. 12) {124} Подпрограма е част от цялата програма, която извършва само дадената на нея задача от цялата програма

296. 398. Какво съответства на подпрограмите в езиците от високо равнище?

296. 1) {102} Пост процеси
296. 2) {105} Функциите
296. 3) {105} Съответстват готови алгоритми и Методи за извършване на дадена операция
296. 4) {118} Подпрограмите в езиците от по високо равнище съответстват на математически операции
296. 5) {123} Подпрограмите улесняват писането на програмата като я разделят на няколко подпрограми и правят програмата по-лесна за разбиране

297. 399. Какви са основните предимства на подпрограмите?

297. 1) {105} Под програмите помагат за изпълнение на различни действия и изпълнението на задачи което прави основната програма по универсална
297. 2) {105} ПП дават възможност да се прави многократно извикване на кода им в програмата.
297. 3) {105} Правят програмата по-лесно четима, като я разделят на определени от програмиста части, които се викат от главната
297. 4) {115} преизползваеми са
297. 5) {115} По-бързо действие. Могат да бъдат стартирани едновременно няколко от тях и да работят без да се налага да се изчакват една друга.
297. 6) {121} заданията са кратки, точни, ясни и лесни за разбиране)класифициране (в подпрограмите). Те използват по-малко памет и са по ефективни
297. 7) {122} Последователност на МИ, които са определени и се намират на 1 място в оперативната памет
297. 8) {124} Подпрограмите позволяват да се пише по-чист код т. е. не е нужно всеки път да пишем една и съща програма по няколко пъти.
297. 9) {125} Многократно преизползване на код. не е нужно записване на две еднакви логически действия два пъти, а само веднъж

298. 400. При кои видове ЕП се проявяват предимствата на подпрограмите?

298. 1) {103} При езиците за програмиране от високо равнище.
298. 2) {105} В езиците за програмиране от високо равнище, подпрограмите са един вид функции, които връщат и някакъв резултат, който обикновено се именова като и самата функция (или може да се нарича процедура). Предимствата на подпрограмите са , че те съдържат всички данни да изпълнят отделно процедура, част от основната програма, като по този начин я олекотяват.
298. 3) {105} Проявяват се при езиците за програмиране от високо равнище
298. 4) {110} При ЕП от ниско ниво.
298. 5) {111} Функционалните ЕП, Обектно ориентираните ЕП.
298. 6) {111, 116} При езиците за програмиране от високо равнище
298. 7) {115, 115} При езиците за програмиране от високо равнище.
298. 8) {121} Езиците от ниско ниво.
298. 9) {121} Предимствата на всеки език е различен спрямо друг някойи са значително опростени за написване, но нямат приложение във операционни системи

Други имат високо приложение но са със малко по-сложен синтаксис въпреки това е за предпочитане.

Примерно java има предимството че е достъпен за значително повече устройства спрямо Паскал, при Си се работи по-лесно но за сметка на това неподдържа някойи от по-новите операционни системи.
298. 10) {121} При езиците от ниско ниво; те могат да бъдат контролирани по лесно.
298. 11) {123} При обектно-ориентираните езици за програмиране подпрограмите имат най-голямо предимство от използването им.

299. 401. Кога предимствата на подпрограмите се проявяват най-ярко?

299. 1) {102} В изпълнението на програмата.

299. 2) {105} Предимствата на подпрограмите се проявяват, когато задачата е разработване на голяма програма. Подпрограмите облекчават написването на програмния код.
299. 3) {107} Когато се използват променливи в подпрограмите
299. 4) {111} Проявяват се най ярко **в използваните параметри**
299. 5) {113} Предимствата на подпрограмите се проявяват най-ярко при наличието на големи количества информация.
299. 6) {113} При обработката на абстрактни данни
299. 7) {119} При съставянето на големи програми от много програмисти.
299. 8) {121} При модулното програмиране – групи от често използвани подпрограми се обединяват в модули, което води до по-голяма преизползваемост на кода.
299. 9) {122} предимствата на подпрограмите се проявяват най-добре при необходимост от отстраняване на грешки, при запис на програмата. Например дебъгера служи за отстраняване на грешките при пробване на програмата.
299. 10) {124} При големи програми с условия
299. 11) {124} При работем на ниско ниво.
299. 12) {124} 1 Обема на програмата се намалява. 2)отделните частни задачи се обработват и определят от подпрограмата с ясно и точно описани взаимни връзки.
299. 13) {124} Когато извикваме дадена подпрограма, за да спестим писането ѝ
299. 14) {126} ППГса отделни компоненти (модули) или най-просто съставни части на една Програма Предимствата на ППГ се проявяват най-ярко при

300. 402. Какво представляват формалните параметри в ЕП?

300. 1) {103} Параметри които съществуват без да бъдат задавани.
300. 2) {104} Запазени думи от ЕП нужни при писането на програмен код
300. 3) {105} **Формалните параметри се използват при дефиницията** (началното задаване) **на параметри**.
300. 4) {106} това са формално зададени данни.
300. 5) {110} Адреси в ОП.
300. 6) {111} Извикват се при изпълнението на функция, като имат един допълнителен параметър, който се отъждествява с тяхното име.
300. 7) {121} Параметри, които по подразбиране се създават в процесора.

301. 403. Какво представляват фактическите параметри в ЕП?

301. 1) {105} Това е стойността на параметъра, след като приеме от външната програма/подпрограма.
301. 2) {105} Фактическите параметри в ЕП представляват реални стойности подадени на метода (ППГ) за изчисление.
301. 3) {110} Фактическите параметри в ЕП са променливи.
301. 4) {113} Променливи, функции, стойност, имена
301. 5) {121} Особен вид процедури с един изходен параметър, който се отъждествява с тяхното име
301. 6) {122} те са параметри който в програмата придобиват реални стойности при изпълнението на програмата.
301. 7) {124} Входни данни.

302. 404. Какво представляват функциите в ЕПВР?

302. 1) {104} бързо и леко писане, не трябват понятия за КС.
302. 2) {106} **функциите** в ЕПВР **са инструкции**
302. 3) {112} Набор от инструкции елементарни операции за извършването на някаква работа. Алгоритъм е съвкупност от дадени елементарни операции за извършването на дадена работа.
302. 4) {115} Бързо и лесно писане, понятен за хората осигурява някои АСД, привични термини.
302. 5) {121} Подпрограми, които за разлика от процедурите, връщат стойност.
302. 6) {121} Функциите в ЕПВР представляват способ чрез който се извършват дадени действие

302. 7) {122} Функциите са множества от МИ, които могат да бъдат извикани на повече от 1 място
302. 8) {124} Функциите в ЕПВР представляват предварително обособени алберит за изпълнение на дадена задача които са дефинирани с еднозначно, име и могат да се изскват в определени битове входни данни
302. 9) {125} Основен вид процедури с 1 изходен параметър, който се отъждествява с техното име
302. 10) {125} Представляват подпрограми (ППГ).

303. 405. Какви видове параметри (поне 4) предлагат езиците от високо равнище?

303. 1) {110} видовете параметри са
параметър псевдоним
формален параметър
фактически параметър
303. 2) {126} Променливи, константи, масиви, асоциативни масиви

304. 406. Какви видове параметри съществуват на равнище МЕ?

304. 1) {102} Формални и фактически
304. 2) {105} Параметър стойност, параметър **променлива**.
304. 3) {109} Параметри
– **стойност**
– променлива
– процедура
– функция
– име
– резултат
– **стойност**

305. 409. Какво представлява съглашението за използване на фиксирани регистри на ЦП при предаване на параметри между подпрограми?

305. 1) {103} За предаване на параметри между подпрограме се използват три системи:
1) фиксирани регистри на централния процесор
2) област от паметта (адресация по база)
3) В стек за параметри (рекурсия)
305. 2) {106} Идеята е да се използва еднакъв (един и същи,) код, за да може те да са съвместими. Параметрите на подпрограмите са еднакви и те използват фиксираните регистри и с цел по-бърза и лесна обработка на заявките.

306. 410А. Какви са предимствата при предаване на параметри във фиксирани регистри на ЦП?

306. 1) {104} По-бърза реализация. По-малко грешки.
306. 2) {115} Знаем точно къде се намират тези параметри в ЦП.

307. 411. Какво се практикува когато регистрите на ЦП са недостатъчни за предаване на всички параметри?

307. 1) {105} Най-удачно е да се направи разширение на регистрите.
307. 2) {111} Параметрите се поместват в област от оперативната памет, свързана с извикващата програма.
307. 3) {111} Допълват се с клетки на голямо ОП
307. 4) {111} използват се флагове
307. 5) {113} Допълват се с клетки на ОП.
307. 6) {115} Параметрите се поместват в област от ОП, свързана с извикващата програма, а на ППГ в базов регистър се предава началния адрес на тази област. Днес най-често тази област в ОП е стек!
307. 7) {123} **Параметрите се изпълняват** по важност когато някой от регистрите се освободи.

307. 8) {124} Допълват се клетки на ОП.
307. 9) {125} параметрите се поместват в област от ОП, свързана с извикващата програма, а на ППГ в базов регистър се придава началният адрес на тази област. Днес най-често тази област в ОП и стек
307. 10) {126} параметрите се поместват в област на ОП, свързана с извикващата програма, а на ППГ в базов регистър се предава началния адрес на тази област. Най-често тази област е стек

308. 412. Кой параметър и днес се предава във фиксиран регистър на ЦП? Защо?

308. 1) {115} Указателят към стека, защото няма как да се зададе параметър на стека.
308. 2) {116} Указателят към стека, защото няма начин как да се зададе началото на стека.
308. 3) {121} променлива,
308. 4) {124} Акумулаторните процесори са кондензатори или транзистори свързани по определена схема
308. 5) {126} Указателя към стека, защото няма как да се зададе началото. на стека.

309. 417. Защо днес предаването на параметри към ППГ най-често се осъществява чрез стек?

309. 1) {102} Предаването на параметри към под-програми чрез стек е по-ефективно, по-бързо, и в по-голямо количество.
309. 2) {102} Защото е най-практично чрез стек.
309. 3) {106} Най-надежден – избягва се риска от зацикляне на програмата. Лесно достъпен. **Евтин**.
309. 4) {106} Защото предаването на параметрите чрез стека е линейно.
309. 5) {121} Поради увеличения обем на информация.
309. 6) {121} защото предаването на параметри чрез стек е по-бързо.
309. 7) {122} Защото е най-удобно тъй като се предполага че последно подадените параметри ще се използват наново
309. 8) {124} Защото стекът е по-удобен и по-уместен начин за предаване на параметри към ППГ.
309. 9) {126} Тъй като са прекалено големи.

310. 418. Какви добавки в архитектурата на ЦП облекчават използването на стек за предаване на параметри?

310. 1) {103} Добавките в архитектурата на ЦП, облекчават използването на стек за предаване на параметри, защото те се фиксират на мястото, което използва ППГ, вместо системния УС, се използва друг регистър, наречен Указател на Кадър (УК) в стека. Така УС може да се използва свободно за временно съхраняване на работни данни.
310. 2) {103} Добавя се указател на кадър за да може указателя към стека да се използва за други изчисления

311. 419. Какво представлява понятието „указател на кадър в стека“?

311. 1) {111} Указател на кадър в стека представлява указател за адреса и реда на изпълнение на кадрите в стека.
311. 2) {111} Определя се мястото на програмата в оперативната памет
311. 3) {113} **Специален списък**, който показва местоположението на дадена програма и показва извършваните процеси
311. 4) {114} Специален регистър, който посочва мястото в стека, което изпълнява програма
311. 5) {124} Указателят на кадър в стека, указва адресът на кавъра.
311. 6) {125} За да се фиксира мястото което използва подпрограмата вместо системния указател.
311. 7) {125} Адресът на даден обект, намиращ се в стека

312. 420. Защо достъпът до параметри, предавани в стек, не се реализира чрез указателя на системния стек?

312. 1) {111} За да може **системния стек да участва в изчислението на междинни данни**.

312. 2) {111} При достъп да стек винаги се достъпва последния вкаран елемент
312. 3) {119} Изграждат се локални променливи за дадения стек.
312. 4) {122} За да се фиксира мястото, което използва програмата, вместо системния указател на стек се използва друг регистър, наречен указател на кадър в стека.

313. 423. Само за предаване на параметри ли се използва апаратният стек на ЦП? Ако да – защо, ако не – за какво друго се използва той?

313. 1) {109} Да, защото след операцията резултатът се записва в горната част на стека заради по-бързо действие.
313. 2) {114} Да. И за временно съхранение на параметри
313. 3) {119} Да защото се използва за адресиране на паметта и ПУ.
313. 4) {121} Използва се и за съхраняване стойностите на операндите
313. 5) {121} пр: f(пр, действ) → записва в пр. резултатът от извършеното деиств. в/у пр.
313. 6) {124} Да. Управлението на устройствата се извършва от УУ (управляващото устройство).

314. 427. Какво предвижда схемата на Си за предаване на параметри чрез стек? Защо?

314. 1) {105} Почистване на стека с цел избягване на задръстване и забавяне
314. 2) {111} Всяка програма сама да почиства стека преди възврат Защото подпрограмите винаги са с фиксиран брой параметри
314. 3) {113} Стекът се почиства от активиращите след възврат от програмата
314. 4) {113} 1) Изпълнява се подпрограма със свой кадър в стека
2) Параметри – запазва се място за тях
3) Записват се входните.
4) Активира се програмата.
5) Запазва се стария указател на кадър
6) Създава се нов кадър – указател на кадър: указател на стек
7) Запазва се място за локални променливи.
8) подпрограмата приключва работа.
9) Освобождава се използваният стек, указател на стек указател на кадър
10) Получава се адресът на възврат от подпрограмата
11) Продължава програмати
314. 5) {115} Първи влязъл, последен излязъл Спомага за линейното изпълнение на програмата.
314. 6) {121} Предаването на параметри чрез стек става по-бързо и лесно и заетата памет е по-малка.
314. 7) {121} Данните се организират по следния начин: който пръв влезе в стека, пръв излиза. Така последния влязал параметър винаги ще се намира на върха на стека.
314. 8) {122} Всяка подпрограма сама да почиства стека преди възврат защото подпрограмите са с фиксиран брой параметри
314. 9) {124} Схемата на Си предвижда да има указател, които да сочи към върха на стека.
314. 10) {124} С звезда (*) след името на параметъра

315. 433. Какво е предназначението на ПУ?

315. 1) {116} подпомагат изпълнението на работата на главните устройства – центр. процесор и ОП

316. 434. Каква е основната разлика между главните и спомагателните устройства на един компютър?

316. 1) {104} Без главните не може да работи а без спомагателните може
316. 2) {105} Главните устройства обработват информациети. а спомагателните е въвеждат.
316. 3) {105} Главните устройства са – **дънна платка**, хард диска, ЦП и ром памет – те извършват изчисленията и съхраняват данните спомагателните устройства подпомагат работата и извеждат обработената от главните устройства информация – монитор, принтер скенер и т. н.
316. 4) {106} Главните устройства са **дънна платка**, процесор, ОП Спомагателните са принтери, скенери и всякакво периферни устройства Разликата, е че без първите процеса на работа

на компютъра не може да бъде започнат. Вторите спомагат за по-добрата му и качествена работа

316. 5) {110} Без главните компоненти не би могъл да работи

316. 6) {115} Главните устройства (процесор и оперативна памет) са абсолютно задължителни за изграждането на един компютър. Спомагателните устройства (входни и изходни) могат да са всякакъв брой и съвсем да липсват, и служат за комуникация с външния свят.

316. 7) {123} Без главните у-ва не може да се използва, а второстепенните може те са по наш избор.

317. 435. Какво е следствието от основната разлика между главните и спомагателните устройства на компютрите?

317. 1) {106} Следва, че с главните устройства процесорът по-бързо извършва действията.

317. 2) {110} Спомагателните устройства работят с десетична бр. с., докато главните в **двуична**, затова е необходимо превеждане от десетична към двуична бл. с. с цел разбира-преобразуване не на информацията която предаваме от спомагателните към гл. устройства.

317. 3) {110} Следствието е, че главните устройства са задължителни при работа с компютър, а спомагателните са опционални.

317. 4) {112} Следствието от разликата м/у главните и спомаг. устройства е че без главните устр. компютърната система неможе да работи а без спомаг. може

317. 5) {115} Устройствата променяни най-много в компютърните системи са ПУ (периферните у-ва), ката промяната най-често е във вход-изход.

317. 6) {116} При интегрирането на УУ на TVоставя място за реализирането на разширени АГУ плаваща запетая.

317. 7) {117} За вътрешна употреба, на процесора при завършване на изчисленията.

317. 8) {117} Спомагателните устройства са за улеснение на потребителя, а главните за обработка на данни и прочие

318. 436. Какви видове ПУ познавате?

318. 1) {102} Видовете ПУ са **външни ПУ и вътрешни ПУ**.

318. 2) {112} Клавиатура, скенер, екран, принтер, модем.

318. 3) {115} Вход/изход,

318. 4) {119} Видове ПУ: – Клавиатура
– скенер
– принтер
– модем

319. 437. Има ли ПУ, които могат да работят както като входни, така и като изходни? Ако не – защо, ако да – как се наричат тези устройства?

319. 1) {106} Периферни устройства които работят като (вх/изх), (I/O) има много, най-често това са външна памет тип, и се наричат „портове“.

320. 438. Какво представлява външната памет на компютрите?

320. 1) {102} Платки.

320. 2) {106} Външната памет е енергоНезависимата памет на компютъра. Тя се използва, защото ОП е енергозависима и при изключване се губи цялото съдържание. **[И когато работи на магнитен принцип ли?]** Също така в ОП не може да се съхранява всички програми, които ЦП някога ще изпълни

320. 3) {110} Оперативната памет неможе да има всички програми които съществуват. Тук се намесва външната памет (в днешните персонални компютри – хард диск). Тя спомага за запазване на информация, но е по-бавна от ОП.

320. 4) {114} периферно устройство

320. 5) {114} Външната памет на компютрите са периферни устройства, които могат да се прикачват към тях и да съхраняват информация
320. 6) {114} Кеш-памет – може да бъде и външна и вътрешна. Външната памет е енергозависима
320. 7) {115} Външната памет се използва за записване на програма, за да може да се освободи оперативната памет

321. 439. Защо е необходима външна памет в компютрите (поне 2)?

321. 1) {104} – защото ОП не е достатъчна
– понякога в ОП не може да се разположи повече от една програма.
– защото е енергонезависима
– дава допълнителни възможности за работа
321. 2) {105} За да можем да съхраняваме повече информация. За да работи по-бързо компютъра.
321. 3) {108} ЦП има пряк достъп до файлове. Повече място за съхранение.
321. 4) {120} Защото вътрешната памет е малка по обем и се използва за да се зареждат програмите в нея

322. 440. По какви показатели (поне 3) се различават запомнящите ПУ?

322. 1) {104} Запомнящите периферни устройства се различават **по оперативната памет**
322. 2) {105} Запомнящите устройства се различават по:
– начин на запомняне
– механизъм за изработка
– **сменяемост на** запомнящото **устройство**
322. 3) {105} 1. Използваната технология зя съхраняване на данните например харддиск (HDD) и Flash
2. Скорост на четене и писане
3. Обем на информацията която можем да запишем.
322. 4) {110} Механична част, Управляваща част, Входно-изходна част
322. 5) {110} Размер (обем на паметта)
322. 6) {114} адресна идентификация, даннова транспорт на данни, управляваща – команди и заявки

323. 441A. Посочете поне две предимства на магнитната в сравнение с оптическата технология за помнене.

323. 1) {105} По-евтини и по-лесни за произвеждане (все още няма оптични памети). Оптичните памети от своя страна са с по-голям капацитет, по-бързи са и вероятно по-надеждни.
323. 2) {110} По-евтина; значително по-издържлива във времето.
323. 3) {116} Оптическата технология за запаметяване е техн. на бъдещето, все още не е в употреба.

324. 441B. Посочете поне два недостатъка на магнитната в сравнение с оптическата технология за помнене.

324. 1) {102} Магнитната технология е значителна по-малка като обем, от оптичната, бързодействието и е по-ниско и обема на данните, които могат да се прехвърлят е по-малък
324. 2) {115} По-неточна; по-несигурна за съхранение във времето

325. 441C. Посочете поне две предимства на оптическата в сравнение с магнитната технология за помнене.

325. 1) {104} По надежна, по голяма от предходните магнитни устройства (MB)
325. 2) {109} По висока скорост на четене и записване. Липса на механично въртящи се елементи при оптичната технология

326. 442A. Посочете две предимства на лентовите в сравнение с дисковите запомнящи устройства.

326. 1) {102} Лентовите запомнящи устр. са изградени от евтин и гъвкав материал, удобни за пренасяне

327. 442B. Посочете два недостатъка на лентовите в сравнение с дисковите запомнящи устройства.

327. 1) {103} ① Лентовите са по-големи физически, но могат да пазят по-малък обем информация
② Лентовите се износват по-бързо и в следствие натова има загуба на качеството.

328. 442C. Посочете две предимства на дисковите в сравнение с лентовите запомнящи устройства.

328. 1) {104} Не са енергозависими. По-голям обем.
328. 2) {105} Дисковите устройства за запаметяване могат да съхраняват по-голям обем информация, данните могат да се променят, по-компактни са, изработката не е скъпа.
328. 3) {105} По-голям обем на записваната информация и бързина на запис
328. 4) {106} Дискови устройства – те са с по-сложна механика, но за сметка на това се понижава опасността от загуба на носителя, обемът който могат да съхраняват е по-голям.
328. 5) {106} По-голяма памет, по-разпространени в днешни дни

329. 443A. Посочете поне две предимства на твърдите в сравнение с гъвкавите магнитни дискове.

329. 1) {119} а) Твърдите дискове пазят информацията по-дълго.

330. 443B. Посочете поне два недостатъка на твърдите в сравнение с гъвкавите магнитни дискове.

330. 1) {104} по-скъпи; лесно чупливи

331. 444B. Посочете поне два недостатъка на устройствата със сменяем в сравнение с устройствата с несменяем носител.

331. 1) {105} Тези със сменяе4м носител са с неограничен капацитет. Сменяемият носител може да бъде местен.
331. 2) {105} Устройствата със сменяем носител има по-голям риск от загуба на информация
331. 3) {105} Неудобство при работа като допълнителни действия
331. 4) {106} Загуба на данни

332. 444C. Посочете поне две предимства на устройствата с несменяем в сравнение с устройствата със сменяем носител.

332. 1) {105} Сменяем носител – лесно преносим, по компактен, по издръжлив
Не сменяем – по-бързи, трудно се демонтират не толкова издръжливи
332. 2) {106} Могат да се презаписват, трайни носители на информация, **независими**

333. 448. Какво е предназначението на контролерите?

333. 1) {123} Поради бавното действие на периферните устройства е нужно да се свързват към шината на ЦП чрез посредник, наречен контролер.
333. 2) {123} Приемат машинни инструкции, обработват специфични данни и ги изпращат **[Къде?]**

334. 450. Какво представлява понятието „периферен регистър (порт)“?

334. 1) {114} „Периферен порт“ представлява Порт който ни свързва с периферните устройства, т. е. чрез тези портове компютрите се свързват с околния свят, например оптичната мисика със специални адоптер се свързва към този периферен порт, специализиран изход тома представлява периферния порт.
334. 2) {114} средство за свързване и обмен на данни с периферни устройства

334. 3) {123} Периферен регистър представлява регистърът, който отговаря на входно/изходните операции

335. 452. Какви портове най-общо съдържа един контролер?

335. 1) {102} Контролер съдържа входноизходни портове, комуникационен порт.

336. 457. Какви са особеностите на алгоритмите за общуване с периферията в сравнение с обичайните алгоритми за обработка на данни? Защо?

336. 1) {119} Входните и изходните данни се въвеждат или извеждат чрез периферни устройства, от потребителя. Алгоритъма трябва да има интерфейс за взаимодействие с потребителя и периферията.

337. 458. Какво представляват протоколите за обмен?

337. 1) {103} Протоколите за вход/изход са стандартни като между вход/изход има малка разлика
337. 2) {107} Протоколите за вход/изход **стандртизирани** като м/у входни и изходни има малка разлика.
337. 3) {107} Протокола се използва съвместно с IP протокол като обикновенно се наричат TCP/IP комплект от протоколи

338. 459. Какво представляват драйверите?

338. 1) {105} Драйверите са елемент, който компютъра трябва да има за да може да работи програмата, която искаме.
338. 2) {115} Те са софтуерната част от контролера осигурява посредничество м/у КС и ПУ

339. 460. За какво са необходими прекъсванията на ЦП?

339. 1) {111} Когато в ЦП постъпи втора главна програма , ЦП трябва да я изпълни вместо да чака пасивно сигнала готов.
339. 2) {121} За да се изпълнява последователно МИ
339. 3) {122} Прекъсванията служат за определяне, в случаите когато има заредени в ОП за изпълнение 2 главна програма, кога да се възобнови работата на 1вата.
339. 4) {122} Прекъсванията на ЦП са нужни за да се постигне паралелност.
339. 5) {124} За да може да се изпитва втора основна програма докато се изчаква резултат.
339. 6) {125} Прекъсванията на ЦП са необходими за да може ЦП-то да взаимодейства с ПУ

340. 462. Какви типове прекъсвания познавате?

340. 1) {111} Прекъсване когато в опашката чака друга програма Настоящата чака нов параметър
340. 2) {114} Типове прекъсвания?
340. 3) {124} Цифрово и аналогово прекъсване.
340. 4) {126} Системата за прекъсване служи при зареждане на една програма, при нужда да се възобнови работата на 1вата програма. Видове – системни, програмни, входно-изходни.
340. 5) {126} за входно изходни операции

341. 463. Какво е характерно за немаскируемите прекъсвания?

341. 1) {113} Операцията се извършва по-бързо

342. 464A. Какво представлява понятието „маска на прекъсване“?

342. 1) {121} **Свойство** чрез което можем да прекъснем определени елементи един от друг.

343. 464B. Защо е необходимо понятието „маска на прекъсване“?

343. 1) {111} за да съобщи на програмата, че е била прекъсване
343. 2) {111} Специален управляващ бит, който показва дали в момента ЦП може да прекъсне своята работа за да обработи заявеното прекъсване
343. 3) {123} За да не се спира изпълнението на програмите.
343. 4) {126} забранява изпълнението на възникнали прекъсвания (флаг в процесора)

344. 465. Как се променя основният цикъл на УУ заради прекъсванията на ЦП?

344. 1) {111} Програмния брояч се увеличава, според заявките за прекъсване към ЦП
344. 2) {114} За да може ЦП да използва втора главна програма, докато чака резултата от Пу

345. 469В. Защо прекъсванията по вход/изход възникват първи?

345. 1) {113} Защото входно/изходните устройства имат и механични компоненти, което ги прави много по-бавни от ЦП и ОП.
345. 2) {121} Имат по-висок приоритет; пращат заявки към ЦП, които изискват обработка в определен период от време.
345. 3) {122} Защото в един ЦП може да се извършват определен брой операции и когато са повече води до операция прекъсване. Поради тази причини възникват 1ви
345. 4) {122} За да предпазят вътрешни проблеми
345. 5) {123} Защото те са началото и края на една програма (код).
345. 6) {124} Защото периферните устройства са първите, които изназват прекъсванията по вход/изход

346. 470. Какво е особеното при прекъсването за начално установяване (reset)?

346. 1) {111} Губят се данните от адресите в ОП
346. 2) {114} По този начин се дава възможност но ЦП да изпъл. функ. на МИ.
346. 3) {124} Използва се за обхождане, заедно с автоувеличение
346. 4) {124} CMOS прави проверка на системата и в ОП се зареждат програмите необходими за работен
346. 5) {124} При „reset“ оперативната памет от тип RAM, туби съхраняваната информация или програма и трябва да се зареди наново докато ROM не се нуждае от електричество за да помни информация
346. 6) {125} след прекъсването програмата трябва да продължи нормалния си ход.

347. 471. Какво е особеното при програмните прекъсвания?

347. 1) {111} Не е възможно две В/И устройства да заявят прекъсване едновременно. Има два типа прекъсвания – маскируеми и немаскируеми. **[Типичен отговор в стил „И аз съм чувал нещо по въпроса!]**
347. 2) {111} Възникват по време на изпълнение на МИ
347. 3) {121} В зависимост от програмата при прекъсване тя остава в паметта до завършване или се прекратява със загуба на информацията.
347. 4) {122} Програмните прекъсвания възникват по време на изпълнението на МИ.
347. 5) {124} Програмата след прекъсването трябва да може да продължи с нормалния си ход. Прекъсванията се получават, когато ПУ изпращат заявка към ЦП, които заявка той трябва да изпълни като прекъсне за определен период от време текущата програма.

348. 472. Всички видове прекъсвания ли трябва да предоставя един ЦП? Защо?

348. 1) {121} Всички видове от прекъсвания трябва да предоставят един ЦП, когато се търси ефективност на системата;
348. 2) {121, 126} Да, защото ПУ предават заявка за обслужване, която изисква от ЦП временно да спре изпълнението на текущата програма.
348. 3) {124} Да, за да е съвместим с различни видове архитектури
348. 4) {126} Не, защото всеки ЦП е различен.

349. 473. Избройте варианти за възникване на прекъсване по контрол на програмата (поне 3).

349. 1) {121} Грешен регистър

350. 474. Избройте варианти за възникване на прекъсване по контрол на апаратурата.

350. 1) {111} – недействителен код на операция
– грешен код на операция
– опит за делене с 0

350. 2) {121} Липса на електричество, бъг в системата, **не изправност в** хардуера или **софтуера**

351. 476. По какви начини могат да се приоритизират прекъсванията по вход/изход?

351. 1) {111} Програмен, фиксиран или кръгов.

351. 2) {111} Отделните входове, имат отделни маски за забрана на прекъсванията. Освен това, определя се приоритет на ЦП – заявка с по-висок приоритет от този на ЦП се обработва, но иначе – не.

351. 3) {121} Ако една задача отнема прекалено голямо време за изпълнение, тя бива прекъсната и започва изпълнение на друга задача, след което се връща към предишната задача.

351. 4) {121} Тези които са с по-важно значение отколкото други с по маловажно.

351. 5) {125} По ред на извикване на командите.

352. 477. Как се реализира програмен приоритет на прекъсванията по вход/изход?

352. 1) {111} Понеже не е възможно две ПУ да заявят едновременно прекъсване на забранените прекъсвания не се обработват и се натрупват няколко,1 това са така нар. висящи прекъсвания

352. 2) {113} Когато адресната шина е свободна управляващата шина чака сигнал.

352. 3) {121} Понеже не е възможно две ПУ да заявят едновремено прекъсване, забранените прекъсвания не се обработват и се натрупват няколко това са така наречените висящо прекъсвания

352. 4) {121} Когато централния процесор задии може да ресурс за изпълнение на дадена програма той проверява кои програми са в режим чакащи и изпълнява програмата, която е с по-висок приоритет.

352. 5) {122} Реализира се когато има нужда в случай че **трябва да извика вход** за да приеме информацията, която е нужна следователно и да използва изход по същия принцип

352. 6) {123} Процесите който искат да заемат ресурс от процесора са наредени в опашка в ОП зависи от кое архитектурно ниво на операционната система се получават ресурс с приоритет

352. 7) {124} Не е възможно две ПУ да заявят едновременно прекъсване.

352. 8) {125} Понеже не е възможно две ПУ да заявят едновременно прекъсване, забранените прекъсване не се обработват и се натрупват няколко – това е т. нар. висящи прекъсване

353. 478. Какви схеми за програмен приоритет на прекъсванията по вход/изход познавате?

353. 1) {113} Заявка

353. 2) {114} Схемите за програмен приоритет са 2:
1. Задава се приоритет на ЦП и ако инструкцията е с по-голям от този на ЦП, тогава настъпва прекъсване.

353. 3) {123} input, output bin Inait

353. 4) {124} Схеми за програмен приоритет
Няма Как 2 (про) сигнала от програма да бъдат подадени и обработени едновременно. затова вата влиза в стадии възникване според приоритета на сигнала подали е с фива. Процесор спира работа по изчисленияна си , до момента изчислява полученото и се връща там откъдето е прекъснало

353. 5) {124} С 256 степени 8 битов звук достатъчен за човешки говор и 65 536 степени 16 битов кодиране във вид на памет.

354. 479. Как става обхождането на ПУ при фиксиран програмен приоритет?

354. 1) {121} Колкото е голяма думата, толкова са и броя битове

354. 2) {121} Използват няколко адресни клетки. Използват изместване

354. 3) {126} При заявка за прекъсване обхождането винаги става в един и същи ред.

355. 480. Как става обхождането на ПУ при кръгов програмен приоритет?

355. 1) {101} Винаги се започва от едно и също ПУ и всички ПУ се обхождат в еднакъв ред, но след последното следва първото,, т. е. когато бъде намерено и обработена заявка, обхождане продължава оттам, до където е стигнало

355. 2) {101, 122} Варти от 1 до N и после пак започва от 1 до N, но когато обслужи дадено прекъсване започва да проверява от следващото

355. 3) {111} При кръгов програмен приоритет обхождането започва от там откъдето е завършило. Това става при заявка за прекъсване. **[Този отговор е изключително неясно формулиран!]**

355. 4) {111, 113} Винаги се обхожда в един и същи ред от 1 до N и след обслужване на прекъсването отново започва с 1

355. 5) {113} Винаги се започва от едно и също ПУ и винаги ПУ се обхождат в еднакъв ред – от първото към последното и когато беди намерено заявка, тя се обработва след което обхождането започва отначало.

355. 6) {121} Започва с кръгово обхождане.

355. 7) {124} Винаги се започва от едно и също ПУ.

355. 8) {124} То започва от първия (най-горния) елемент и се движи обратно на часовниковата стрелка.

355. 9) {124} Винаги се използва същото ПУ и всички ПУ се обхождат в еднакъв ред.

355. 10) {125} Винаги се използва същото ПУ и всички ПУ се обхождат в еднакъв ред, но след последното следва първото

355. 11) {125} Винаги се започва от едно и също ПУ и всички ПУ се обхождат в еднакъв ред – от първото към последното и когато бъде намерена заявка, тя се обработва. след което обхождането започва от начало.

355. 12) {126} периферните устройства биват обхождани последователно от контролера

356. 483. Какъв е най-простият принцип за автоматизиране на програмирането?

356. 1) {110} Програмата да се извиква в цикъл

357. 484A. Посочете поне два недостатъка на езика Асемблер в сравнение с МЕ.

357. 1) {111} – Програмите се пишат по-бавно
– Необходими са познания за компютърни системи.
– Пълният контрол над комп. с-ма е опасен

357. 2) {112} По-бавен

357. 3) {112} CISC – микропрограмируеми инструкции – сложни МИ са с различна структура, усложняват и работата на процесора.

358. 484D. Посочете поне две предимства на МЕ в сравнение с езика Асемблер.

358. 1) {118} ① пише се по-леко и по-бързо
② Разбираем е за хората

358. 2) {123} МЕ е по-понятен на компютърната система. МЕ позволява достъп до пълното количества ресурси на компютърната система

358. 3) {123} МЕ са по-ефикасни от гледна точка на структорно изпълнение на всеки МЕ може да се характеризира с отделен собствен Асемблер

359. 485. Един единствен език Асемблер ли съществува? Ако да – защо, ако не – по какво съществено се различават отделните езици Асемблер?

359. 1) {123} Да съществува един Асемблер защото той е н език и трябва да бъде един и същ при всички компютри.

359. 2) {124} Няма един единствен език Асемблер. всеки отделен процесор има свой МЕ Има малки разлики в брой на МИ и операторите.

360. 486. Какви са общите принципи (поне 3) на асемблерните езици?

360. 1) {114} Асемблерните езици са езици от ниско ниво, нужно е голямо количество символи за писането на код. Кодът е труден за четене

360. 2) {121} принципи на асемблерните езици
1) използване на двоична иззиз. бройна сестем
2) работа с регистрите на процесора
3) облекчава бързодействието на програмата

360. 3) {122} – всяка операция (команда) се записва на нов ред
– могат да работят с символни адреси, които са по лесни за работата на програмата
– кода е по добре оптимизиран за системата но се пише по бавно
– нужна програма която да подготви кода ди машинен ез. (Пр. при компилация символните адреси се превеждат обратно на цифрови)

360. 4) {124} 1) По-лесни за четене и писане сравнение с машинния език.

360. 5) 2) Всяка операция се записва на отделен ред

360. 6) 3) имат достъп до КС.

360. 7) {124} Ниско почти машинно ниво с лесен но бавен за писане код.

361. 487А. Посочите поне две предимства на езика Асемблер в сравнение с език от високо равнище, който познавате добре (Паскал, Си, ???).

361. 1) {111} бързина (изключване на ненужни операции) на операциите
бързина (достъпа до процесорните регистри е многократно по-бърз от) на достъпа (достъпа до ОП но данните

361. 2) {114} Предимствата на езика Асемблер който е от ниско равнище са: заделя се специално място в паметта при извършване на определените операции

361. 3) {114} Има директен достъп до регистрите на ЦП и до ОП

361. 4) {115} Предимства – разбираем за хората, програмата се пише по-леко и по-бързо

361. 5) {121} **Достъпване** на данни директно по адреси и бързодействие

361. 6) {121} – Едно от предимствата е че можем значително да съкратим кода същевременно да работи и върши същата работа
– Второ имаме по-голямо приложение спрямо някои езици, намиране и разрешава грешките по-бързо

361. 7) {121} ① Може да се работи на равнище най-близко до машинния език. ② може да се използват функции, което не може при езиците от по-голямо равнище

361. 8) {122} Разбираем за хората, пише се по- леки и по-бързо, промяната е лека и без грешки

361. 9) {125} Предимства на Асемблер:
– разбираем за хората/програмиста
– лесен за четене/писане

362. 487В. Посочете поне два недостатъка на езика Асемблер в сравнение с език от високо равнище, който познавате добре (Паскал, Си, ???).

362. 1) {103} изисква програма и време

362. 2) {112} 1. Част от ОП не може да се използва, защото там работи транслаторът от Асемблер
2. Многослойните библиотеки имат недостатъка да са бавни.
3. Използват се всички видове ЦП
4. бавно писане

362. 3) {119} Асемблер достъпни са всички МЕ, Използват се всички видове данни на ЦП.

362. 4) {125} Си използва повече ресурсна ла от асемблера, както и асемблера има директен достъп до машинният код и може чрез него по-лесно да се достигне и кроцамират самитеелектронни компоненти.

363. 487C. Посочите поне две предимства на език от високо равнище, който познавате добре (Паскал, Си, ???) в сравнение с езика Асемблер.

363. 1) {104} Липсва контрол по време на изпълнение **[На какво?]**. По-бавни са. **[Много е сложно да си представим скорост на език!]**

363. 2) {122} Входните променливи се изчистват от програмата.

364. 487D. Посочите поне два недостатъка на език от високо равнище, който познавате добре (Паскал, Си, ???) в сравнение с езика Асемблер.

364. 1) {115} Асемблер:
– бавно писане
– пълният контрол над компютърната система е опасен
– липсва контрол над изпълнението

364. 2) {116} по-сложни,

364. 3) {124} Асемблер при изпълнение изпълнява последователно всеки ред доколкото в C++ например се компилира целият код и го изпълнява наведнъж. Неможем да пишем на компютърен език в C++.

364. 4) {125} Езиците от високо равнище като Си, Паскал, Jeva
– са по-бавни от Асемблер;
– не могат да се оптимизират за специфичен процесор като Асемблер.

365. 488. Опишете структурата на един оператор на Асемблер.

365. 1) {115} транслатор компилатор

365. 2) {115} Какво да се извърши / с какво т. е. 2 части в 1-ва се посочва операция во 2-ра с какво да бъде извършена

366. 491. Какво може да бъде записано в полето за мнемоничен код на операция при езика Асемблер?

366. 1) {111} Имена, константи

366. 2) {111} Код на операция, операнди

366. 3) {125} Текстовият еквивалент на КОП на МИ

367. 494. Как се определя броят на операндите в един оператор на езика Асемблер?

367. 1) {111} Те варират според оператора.

367. 2) {111} **Операндите се разделят** с запетая

367. 3) {113} от това колко дълга програма правим.

367. 4) {113} Определя се от кода на операцията и всеки код на операцията изисква определени операнди.

367. 5) {119} Според първия байт на оператора.

367. 6) {121} Броят на операндите в един оператор на Асемблер е точно четири: поле за изчисления на стойността, поле з0а адрес , на който да бъде записана стойността, поле за самата стойност и поле за допълнителни записки.

367. 7) {121} Опредил се от кода на операцията и всеки код изисква различни операнди

367. 8) {122} На всеки ред в езика Асемблер може да има само по 1 операнд.

367. 9) {123} Броят на операндите в един оператор на езика асемблер се определя от броя на операциите

367. 10) {124} Според количеството памет което заемат за извършването си.

368. 499. Как в езика Асемблер може да бъде посочван размерът на обработваните данни?

368. 1) {111} с ключовите думи BYTE WORD …

368. 2) {114} Съвместната работа и управлението на посочените блокове се извършва от от обработваната информация Два размера 5,25 и 3,5 инча

368. 3) {120} Чрез оператор ptr[](указател към адрес от паметта), като преди него се поставя друг оператор, който определя колко байта да бъдат обработени.

Пример: dword ptr [0AF4B34C] – ще прочете двойна дума (4 байта от адрес 0AF4B34C до 0AF4B34F.

368. 4) {124} име на променлива [размер]
368. 5) {124} Размерът на данните се посочва като се мащабира по броя байтове

369. 501. За какво служат символичните имена в езика Асемблер?

369. 1) {103} Адресното поле на МИ посочва адрес от ОП
369. 2) {103} Адресното поле на Ми посочва адре от ОП
369. 3) {107} За да се облекчи програмата
369. 4) {110} Символичните имена в езика Асемблер са съкращения от думи в латинския език: Пример – test в асемер е TST
369. 5) {111} Символичните имена в езика Асемблер служат за съкращението
369. 6) {113} Стойност на регист променява на добавляване на размер промяни на дани (в брой клетки)
369. 7) {121} Служат за кратко и ясно кодиране
369. 8) {124} Те са само символи които помагат за разпознаването им едни от други.
369. 9) {124} За по-лесно разпознаване
369. 10) {124} За по-удобно задаване на адреси от ОП
369. 11) {125} Те са наименованията на адреси в ОП.

370. 502. Как става определянето (дефинирането) на едно символично име в езика Асемблер?

370. 1) {103} При превода всеки оператор ще бъде трансформиран в МИ или др. Данни разположени някъде в ОП. Символично име се дефинира като се запише в полето за етикет на този оператор.
370. 2) {104} _asm
370. 3) {111} с адрес от оперативната памет
370. 4) {111} Символично име на оператор се записва в полето за етикет на оператора всички асемблерни директи и за данни би трябвало да имат символично име в своето етикетно поле.
370. 5) {114} В кавички <Name>
370. 6) {114} За отразяването на данните се използват 2 схеми. Едната директива и посочване на всеки операнд. Различни директиви за всеки вид данни
370. 7) {115} Символичното име в Асемблер служи, за да улеснява работата на програмиста
370. 8) {120} <име на тия данни> <име на променлива>
370. 9) {121} add 2ax, y
370. 10) {121} Чрез добавяне на символ (или низ) пред адресното поле.
370. 11) {122} Това е символично име на оператор и се записва в полето за етикет на оператора Всички асемблерни данни би трябвало да имат символично име в етикетното си поле
370. 12) {124} Чрез съответната команда в езика Асемблер.
370. 13) {126} Символично име еквивалентно на началния адрес **[Кой начален адрес?]** се дефинира като се записва в полето за етикет на оператора

371. 503. Как стандартно се установява връзката между едно символично име и съответстващия му адрес от ОП при езика Асемблер?

371. 1) {111} чрез адресиране по части
371. 2) {121} При компилиране транслаторът на асемблер заменя символичните имена със съответстващите им адреси в ОП {121} Цената на ОП намалява наполовина приблизително на всеки 3 години.
371. 3) {121} чрез регистрите
371. 4) {122} При компилация символичните имена се заместват от съответстващия им адрес от ОП
371. 5) {122} Символично име на оператор се записва в полето за етикет на оператора. Всички асемблерни директиви за данни би трябвало да имат символично име в своето етикетно по-

ле. При компилация, символичните имена се заместват с адреса от ОП, който те представят.

371. 6) {124} Чрез регистра

371. 7) {126} Символично име на оператор се записва в полето за етикет на оператора. Всички Асемблерни директиви за данни би трябвало да имат символично име в своето етикетно поле. При компилация символичните имена се заместват с адреса от ОП, които те представляват.

372. 504. Къде в езика Асемблер се използват символичните имена?

372. 1) {102} Символичните имена в Асемблер се използват за задаването на параметри за по-лесна разпознаваемост от програмиста при по-нататъшна употреба

372. 2) {111} Адреси от ОП, които се задават с последователност от символи Имената са еквивалентни с отбелязаните адреси

372. 3) {114} Символичните имена

372. 4) {115} При задаване и извикване на параметри

372. 5) {115} За **да облекчат програмата** адресите на ОП, които **представляват се отбелязват** чрез символични имена

372. 6) {115} При механизмите за именоване на адресите.

372. 7) {118} За улесняването на програмиста, тъй като те дават смисъл на даден адрес от ОП

372. 8) {124} Символичните имена в Асемблер се използват като аргументи на еднаците в езика. Имената могат да носят еднозначно или преносно значение Преносното значение има когато името отбелязва стойност.

372. 9) {124} В таблиците, които се попълват от транслатора на първи пас от превода на една програма.

372. 10) {125} използван се за задаване на адреси от оперативната памет

372. 11) {125} Те са наименования на адреси от ОП.

373. 507. В каква бройна система се записват константите в езика Асемблер? Защо?

373. 1) {110} Шестнайсетична бройна система.

373. 2) {114} В естествено константите да се записват първичната за хората десетична БС БС е неудобна при МИ. За логически операции където **двуичния** запис е по ясен

373. 3) {114} записват се в десетична ПБС за улеснение на хората, но при логически операции може да се използва двоична ПБС.

373. 4) {115} двоична – бързодействие

373. 5) {115} В десетична бройна система.

373. 6) {115} в десетична, за по-лесна работа при програмиране

373. 7) {121} Двоична бройна система.

373. 8) {122} Константите се записват в десетична бройна система, защото е по-разбираемо за хората и лесно се преобразува в двоична

373. 9) {124} Шестнайатична бройна система (КЕХ)

373. 10) {125} Двуична бройна система, защото е най-близка до МЕ.

373. 11) {126} В десетична бройна система. Защото е по-лесно за човека да работи десетична бройна система.

374. 508. Как в езика Асемблер може да се посочи основата на използваната бройна система при запис на константа?

374. 1) {114} В езика Асемблер може да се посочи основата на използваната бройна система при запис на константа.

374. 2) {121} Той борави с двоична БС.

375. 509А. Какво представляват изразите в езика Асемблер?

375. 1) {114} Машинни инструкции

376. 509В. Къде в езика Асемблер се използват изрази?

376. 1) {125} ПНавсякъде

377. 510. Каква е разликата между израза A+1 в езика Асемблер и в ЕПВР?

377. 1) {102} При ЕПВР стойността на А започва от 2, а при езика Асемблер от 1.

377. 2) {121} ecx, eux, esi, edi …

регистри с общо предназначение

377. 3) {124} Изразът [A+1] в езика Асемблер посочва адрес, на който сме задали псевдоним А и ни дава адресът изместен с –. При ЕПВР това би било стойност на променливата А,+1.

377. 4) {125} Изисква прочетената от ОП стойност на променлива да се увеличи с 1.

378. 520А. Защо основната отличителна черта на програмирането с език от тип Асемблер е дългият текст на програмата?

378. 1) {111} Защото езикът Асемблер е близък до машинния език (МЕ).

378. 2) {111} Работа с **лимитен** брои регистри

378. 3) {113} Поради нуждата от изписване на всяко едно действие подробно и с правилната адресация.

378. 4) {121} За сметка на опростяването на кода, той става по-голям като обем

378. 5) {121} пряк Достъп до частите на компютъра програмите се свеждат до ниско ниво и се изпълняват от ЦП.

378. 6) {121} Защото Асемблер работи с регистри и позволява обстойно описване на програмата.

378. 7) {124} Защото за разлика от езиците за програмиране от високо ниво, Асемблер е машинен език. Всяка команда има едно определено действие. Командите извършва действия директно в кода на компютъра

378. 8) {124} Защото всяка операция се записва на отделен ред.

378. 9) {124} Защото се пише на ниско, почти машинно ниво.

378. 10) {124} Защото Асемблер е език от нисък клас, които позволява пълен достъп до компютъра Основното предимство на Асемблер е скоростта (performance(е), кодът е голям по размер защото всяка операция се описва пълно и няма никаква вътрешна смущение (оптимизация и/или компилация)

379. 520В. Каква е основната отличителна черта на програмирането с език от тип Асемблер?

379. 1) {104} Пише се на един изключително близък до машиния език

379. 2) {104, 111} Те са частен случай на работа с обикновени текстове

379. 3) {111} **Езиците** от тип Асемблер **са частен случай на работа с** обикновени **текстове**.

379. 4) {111} При програмиране с език от тип Асемблер се програмират директно машинните инструкции, които да се изпълнят от ЦП.

379. 5) {114} представляват макро апарата и са езици от ниско ниво

379. 6) {114} Правилата, които определя как става дефинирането и как сползвасе на съкраштаване на текста.

379. 7) {121} Предимството е, че е достъпен целия МЕ, пълен контрол над изпълнението, използва всички видове данни от ЦП.

379. 8) {122} Те са случай на работа с обикновенни текстове.

379. 9) {122} Имат съответствие 1:1 с МЕ. Те са частен случай на работа с обикновени текстове.

379. 10) {123} Асемблер е ЕП от ниско ниво и трябва да се внимава с адресирането и освобождаването на памет.

379. 11) {124, 125} Те са частен случай на работа с обикновени текстове

379. 12) {124} Всяка операция се извършва с повече от едно действие, всичко се пише по-подробно.

379. 13) {125} Това е частен случай на работа с обикновени текстове.

379. 14) {126} Всеки процесор е различен, следователно има и свой собствен език Асемблер. За разлика от МЕ, езикът Асемблер е разбираем за хората. Програмирането с език Асемблер е

програмиране от ниско ниво, едно над програмирането на МЕ. Програмирането с език Асемблер е по-лесно и разбираемо, по-лесно се отстраняват грешки отколкото при МЕ.

380. 521. Какво използват хората, когато трябва бързо да създадат дълъг текст?

380. 1) {121} Периферни устройства за печат.
380. 2) {121} Хората използват предварително подготвена информация.
380. 3) {123} Използват се програми за бързо въвеждане
380. 4) {123} За бързо написване на дълги текстове използват текста генератор

381. 522. Какво представлява макроапаратът, включен в някои ЕП?

381. 1) {121} Макроапаратът служи за управление на съкращенията в текста на програмния код.

382. 523. Свързан ли е макроапаратът с програмирането? Ако да – как, ако не – защо?

382. 1) {101} Да, защото за неговото осъществяване е необходима специфична програма, която да заменя съкращенията с пълната дума Този процес се нарича макрогенерация
382. 2) {101} Да. Макроапарата определя как става дефинирането и използването на съкращения.
382. 3) {111} Да, той определя как става дефинирането на съкращения.
382. 4) {113} Да, необходима е специална програма, която заменя съкращенията с думи.
382. 5) {113} Макроапаратът определя как става дефинирането и използването на съкращения.
382. 6) {114} Да защото при неговото свързване е необходима специфична програма която да заменя съкращенията с пълни думи
382. 7) {121} Да
382. 8) {121} Улеснява програмирането с използването на съкращения.
382. 9) {122} Да. Свързан е с Асемблер
382. 10) {122} Да, свързан е. Необходима е специфична програма която да заменя съкрашенията с пълните думи.
382. 11) {122} Да, защото за неговото осъществяване е необходимо специфична програма, която да замена съкр. с цеби думе (пълните думи)
382. 12) {125} Макроапаратът е свързан с програмирането. „MACRO“ – начало, „MEND“ край
382. 13) {126} Да, защото благодарение на негомогат да се използват съкращения

383. 524A. Кои езици за програмиране най-често предоставят макроапарат?

383. 1) {104} Обектно ориентираните езици за програмиране предоставят най-често макроапарат.
383. 2) {113} AOA, C#, C++, C
383. 3) {121} Езиците от високо равнище
383. 4) {122} Макроасемблерите – целта е да могат да излзват съкращение
383. 5) {122} Макроапарат предоставят езиците за програмиране от високо ниво.
383. 6) {124} C++, Паскал
383. 7) {124} C++, Assembler, C. PL1 – защото предлагат улеснение като съкращават записа
383. 8) {125} Асемблер, C++, C, PL1 предлагат улеснение като съкращават записа

384. 524B. Защо езиците за програмиране от тип Асемблер най-често предоставят макроапарат?

384. 1) {111} съкращават обема на програмата
384. 2) {111} Те предоставят Макроасемблер с цел да се използват съкращения
384. 3) {121} Заменя съкращенията с пълните думи
384. 4) {121} За удобство и по-лека работа
384. 5) {124} Защото тай съдържа цялата програма

385. 525. Какво представляват Макроасемблерите?

385. 1) {124} Това са асемблери които обхващат по-голям диапазон от функции от нормалните и могат да използват повече и по-сложни команди.
385. 2) {124} работа с макросите (адресите) на ОП

386. 526. Какво представлява макродефиницията?

386. 1) {103} Дефиниция на съкращение означава началото на дефиницията
386. 2) {126} Използване на съкращение.

387. 527. Какво представлява макроизвикването?

387. 1) {122} Кратко и бързо извикване
387. 2) {125} Макроизвиквнето е получения след замяна текст.

388. 528. Какво представлява макроразширението?

388. 1) {113} пълният текст който се използва за съкращението
388. 2) {121} заменянето на макросите в кода с резултатът от съответната макродефиниция.

389. 530. Каква е ползата от наличие на Макроапарат в един ЕП?

389. 1) {103} Бърз превод на програма от друг език.
389. 2) {103, 103} Чрез него се съкращава програмния код.
389. 3) {113} данните стават по-четливи
389. 4) {121} Макроапаратът служи за **олесняване** на програмистите. Съвкупност от инструкции за съкращения на изрази служи за по-бързо писане на текст.
389. 5) {124} използва се за улесняване на множество на брой процеси по 1 и също време.
389. 6) {125} Позволаво чост от написаното да бъде третирано, като коментар.

390. 531. Какво представлява апаратът за условна транслация, включен в някои ЕП?

390. 1) {104} Този апарат премества битове информация с предварително зададен адрес на транслирането.
390. 2) {113} Част от написаното да бъде считано за коментар.
390. 3) {121} Изисква се условие, за да се извърши действието.
390. 4) {122} Проверява някакво условие преди да се извърши дадено действие

391. 533. Каква е ползата от наличие на апарат за условна транслация?

391. 1) {103, 123} Когато изчислената по време на превода стойност отговаря на условието, редовете до ENDC се превеждат в противен случай се игнорират
391. 2) {103} Че при правилните обстоятелства транслацията се изпълнява автоматично.
391. 3) {103} Позволява част от написаното да бъде третирано като условна транслация
391. 4) {111, 113} Когато изчислената по време на превода стойност отговаря на условието, редовете до ENDC се превеждат в противен случай тези редове се игнорират
391. 5) {111, 111, 124, 124, 125, 125} Позволява част от написаното да бъде третирано като коментар.
391. 6) {122} Проверява някакво условие преди да се извърши действие.
391. 7) {122} означава някой от участъците по такъв начин че при определени условия текстът в тях да се счита за коментар.
391. 8) {123} Условното привеждане на код в машинен език.
391. 9) {125} Дава възможност да се създават по-гъвкави програми
391. 10) {126} Когато изчислената по време на превода стойност отговаря на условието, редовете до ENDC се превеждат. В противен случай те се игнорират.
391. 11) {126} Позволява част от написаното да бъде считано за коментар.

392. 534. Какво е предназначението на транслатора от езика Асемблер?

392. 1) {114} включва таблицата на символичните имена, текста на Асемблер и текста на МЕ
392. 2) {114} Универсални компютри
392. 3) {115} Да преобразува информацията

393. 535A. От какъв тип – интерпретативен или компилативен, е транслаторът от асемблер?

393. 1) {110} Интерпретативен
393. 2) {119} Компилаторът от асемблер е от интерпретативен тип.

394. 535B. Защо транслаторът от асемблер е от компилативен тип?

394. 1) {107} Транслаторът от Асемблер е от компилативен тип, защото той превежда (обработва цялата програма, а не прекъсва, когато открие грешка
394. 2) {114} Съобщава програмата и я превежда на МЕ
394. 3) {114} За да може МИ да се изпълняват (и компилира)
394. 4) {115} Защото трябва целият код да се компилира за да има изпълнение.

395. 536. Какви са входните данни на транслаторите от Асемблер и Макроасемблер?

395. 1) {111} **Асемблерин** код без съкращение
395. 2) {113} Машинен език,
395. 3) {115} Адресни полета и данни
395. 4) {124} Параметри за Асемблер и за Асемблер с макроапарат.
395. 5) {124} Входните данни са под формата на текст.
395. 6) {125} – началните адреси
– модули
– стартовият адрес

396. 537. Какво представляват библиотеките с макродефиниции и защо се използват?

396. 1) {107} Обобщаването на дадено множество на едно място се нарича библиотека
396. 2) {112} Това са библиотеки, които съдържат в себе си функции, константи и дават възможност да бъдат използвани в програмата.
396. 3) {115} това са библиотеки с макроси за често изпълняеми подпрограми

397. 539. Защо транслаторът от Асемблер не блокира изпълнението на създадената обектна програма при откриване на грешки в текста на изходната програма по подобие на компилаторите от ЕПВР?

397. 1) {123} Защото е език от ниско ниво и не разполага с необходимата функционалност да блокира изпълнението на създадената обектна програма.

398. 540. Какво включва листингът, получаван като изход от работата на транслатора от Асемблер? Защо?

398. 1) {115} **Текста на изпълнение** на МЕ – Асемблер

399. 543. Каква е реакцията на транслатора от Асемблер при откриване на повторна дефиниция на символично име?

399. 1) {102} Връща грешка и спира построяването на програмата.
399. 2) {102} запълва > с 0
399. 3) {118} Символичното име се дефинира чрез етикета.

400. 547. Какви таблици поддържа транслаторът от Асемблер за да реализира превод?

400. 1) {111} 1. Мнемонично име 2. Символично име. 3. Ас. директива
400. 2) {114} Мнемонично поле, директива, символично име
400. 3) {115} Асемблер съвпада в съотношение 1:1 с МЕ на ЦП, като в следствие на това за всеки ЦП има разлики в Асемблер. Ной е по-понятен за програмиста
400. 4) {119} – Мнемоничен код
– таблица с директиви
– таблица със символичните имена – адреси
400. 5) {124} мнемонично име – машинен код

400. 6) {125} таблици с протоколи

401. 549. Какво е характерно за таблицата за съответствие между мнемонични и машинни кодове на операции?

401. 1) {115} – Адре3сни полета
– Адресация
– Параметра

401. 2) {115} Съдържа още дължина на МИ брой на адресни полета разрешени адресации и други

401. 3) {115} Съдържа характеристики за КОП, като дължина МИ, броя на адресните полета и разшерени адресации.

402. 550. Какво е характерно за таблицата на съответствие между символичните имена и адресите от ОП?

402. 1) {102} На всеки адрес от оперативната памет се задава символично име, като те се слагат в един общ регистър

403. 551. Каква основна променлива поддържа транслаторът от Асемблер и защо?

403. 1) {107} замества я с n байта 0

403. 2) {123} Транслаторите имат еднакви входни и изходни резултати.

404. 553. Какво е необходимо за да може транслаторът от Асемблер да приключи своята работа с едно преглеждане на програмата?

404. 1) {115} И двете програми работят със символични имена, дефиницията малко може да следва изследването.

405. 555. Какви действия извършва стандартният транслатор от Асемблер по време на своя първи пас?

405. 1) {110} превод на числови данни

405. 2) {114} разчитане на програмата

405. 3) {115} Настройва брояча за разположение на програмата, проверява поредицата от байтове дали е етикет, МИ или директива. Ако разпознае поредицата от байтове увеличава БРП или Коригира ТСИ, ако не разпознае връща съобщение за грешен МнКОП.

406. 556. Какви действия извършва стандартният транслатор от Асемблер по време на своя втори пас?

406. 1) {112} Апаратът за условна транслация предвижда при определени условия част от написаното да бъде третирано като коментар

406. 2) {112} Транслаторът от Асемблер се нуждае от допълнителни указания за своята работа Затова са предвидени асемблерските директиви. Чрез директивата ORG <израз

406. 3) {115} Транслаторът превежда текста написан на МЕ и връща листинг с подробна информация

407. 559. Как работят многопасовите Асемблери?

407. 1) {102} От един пас се преминава към друг

407. 2) {119} Многопасовия асемблер отстранява EQU проблеми. По време на първия пас те запомнят директиви EQU. Сред това следва втори пас.

408. 560. Посочете варианти за съхраняване на получената при превод МП.

408. 1) {111} Във оперативната памет и в кеш-паметта

408. 2) {111} В ~~регистър~~ кеш-паметта или в ОП

408. 3) {125} Получената при превод МП може да бъде съхранена във временната памет, в ром каив постоянната енергонезависима памет

409. 561A. Посочете предимства на съхраняването на получената при превода МП в ОП.

409. 1) {103} бърз достъп до резултатите от следващата програма
409. 2) {104} Бърз достъп до МП по време на изпълнение
409. 3) {113} Съхранението на МП в ОП – по лесно и бързо се достъпва от процесора, действието се извършва по време на изпълнение.
409. 4) {121} **Скоростта** на съхраняване на данните **е бърза**, достъпът до се извършва за константно време, позволява записването на голям брой
409. 5) {122} МП е напълно готова за изпълнение въвеждането на МП в ОП е бързо
409. 6) {124} Предимството да се съхранява МП в ОП е в това, че МП може да бъде прекратена във всяко едно време и заедно с моментното 2 състояние и да бъде извикана отново, като предължи оттам откъдето е прекъсната. Т. е: МП се съхранява в ОП за да може и да бъде прекратена при нужда най естествена при прекъсване.
409. 7) {124} Предимства на МП е, че не се изисква ел. ток за запазване на информацията, също така МП е по-издържлива и затова по-скъпа

410. 562A. Какво представлява зареждащата програма?

410. 1) {103} Програма която задейства стартиращите функции
410. 2) {104} Програма в процес на изчакване.
410. 3) {113} Зареждащата програма представлява алгоритъм от код
410. 4) {121} Зареждащата програма за да се зареди има нужда не само от Постоянна памет а и от Допълнителна памет Преди същата програма да се зареди тя трябва да бъде записана някъде предварително за това са нужни 2 вида памет
410. 5) {122} Програмата създадени във външната памет се нарича зареждаща програма
410. 6) {125} Когато машинната програма се съхранява във външната кеш памет

411. 562B. Кога се използва зареждащата програма?

411. 1) {104} Зареждащата програма се използва при стартирането на самата програма.
411. 2) {111} рекурсивните ППГ не са проблем – това е варианта
411. 3) {113} При стартиране.
411. 4) {121} Когито в последствия на операциите се появи нужда специфичната програма. Примерна BIOS или ОС.
411. 5) {123} При стартиране за пръв път на компютарат или определена програма.
411. 6) {124} Когато трябва да зареди програма.

412. 563. Какво представлява абсолютен обектен код „образ на паметта“?

412. 1) {113} Прекият запис в ОП не е удобен
412. 2) {114} Мн не се записва пряко в ОП
412. 3) {121} МІ се разделя на няколко текстови записа ца да се преодолее **препълване на регистъра**.
412. 4) {121} Абсолютен обектен код е начина на записване на информацията в паметта на компютърната система.
412. 5) {122} Прекият запис в ОП не е удобен В края на 1-ви пас става известен размерът на програмата в ОП и това дава възможност за моделиране на ОП в ВП
412. 6) {123} Прекия запис в ОП не е удобен. В края на първи пас става известен размерът на програмата в ОП
412. 7) {123} Чрез „адресна“ памет.
412. 8) {124} Могат да се използват многократно;
412. 9) {125} Съчетание от знаци и цифри, който указва точното местоположение на първата клетка от паметта, в която е записана данната.

413. 564. Как се реализира зареждането в ОП при абсолютен обектен код „образ на паметта“?

413. 1) {111} Прочитат се **адресът и размерът на зареждане** и след това една единствена операция четене от ВП в ОП

413. 2) {113} Зареждащата програма чете записите, докато намери заглавен запис на исканата програма. След това продължава четенето докато намери стартов запис и прави безусловен преход към адреса, посочен в този запис. След прочитането на поредния текстов запис, се определя колко бита о на какъв адрес в ОП.

413. 3) {121} При образ на паметта програмата се намира във външната памет. При нейното стартиране те се вкарват в ОП на определен или произволен адрес и ЦП я изпълнява.

413. 4) {124} Адресът и размера на зареждането се прочитат и след това четене от ВП в ОП.

414. 565. Какво представлява абсолютен обектен код „поредица от записи“?

414. 1) {121} Поредица от записи събрани в един код.

414. 2) {125} Г този случай се използва част от или двойка регистри на ЦП

414. 3) {125} Ограничения размер на физическия запис във ВП може да бъде преодолян като МП се раздели на няколко текстови записа

414. 4) {126} Чрез Overclock (овърклок).

415. 566. Какви видове записи съдържа абсолютният обектен код във вариант „поредица от записи“?

415. 1) {124} Идеята на стековите процесори е използване на стекови регистри за използване на стек

416. 567. Как се реализира зареждането на програмата в ОП при абсолютен обектен код „поредица от записи“?

416. 1) {111} Машинната програма се разделя на няколко текстови записа.

416. 2) {111} **Прочита се** адреса и **размера на зареждане** и след това една единствена операция четене от ВП в ОП

416. 3) {114} Когато се реализира зареждането на програмата в ОП при абсолютен обектен код „поредица от записи“

416. 4) {122} Ограниченият размер на физическия запис във ВП може да бъде преодолян като МП се раздели на няколко текстови записа: заглавен, текстов и стартов.

416. 5) {122} Свързването на програмата става по 2 начина статично и динамично. При „поредица от записи“ зареждането на програмата се осъществява чрез модули.

417. 568A. Посочете две предимства на варианта „образ на паметта“ в сравнение с варианта „поредица от записи“ за съхраняване на машинните програми.

417. 1) {122} При „образ на паметта“ намирането и изпълнението на програмата е по-лесно и бързо.

417. 2) {123} При варианта „образ на паметта“ има по-бързо достигане до желаната стойност и по-малко заета ОП

417. 3) {124} скорост и консистентност на записа (данните)

418. 568C. Посочете две предимства на варианта„поредица от записи“ в сравнение с варианта „образ на паметта “ за съхраняване на машинните програми.

418. 1) {122} поредицата от записи е бърза, лесно се променя докато в образа на паметта не е така!

418. 2) {123} По лесно намиране на записите, по малко изразходване на ресурси и време за намирането им.

419. 569. Какви са предимствата и недостатъците на абсолютния обектен код?

419. 1) {103} Програмата се пише трудно и бавно и грешките се укриват

420. 569A. Какви са недостатъците на абсолютния обектен код (поне 2)?

420. 1) {121} Голям обем от данни
420. 2) {121} Заемане на повече място

421. 569B. Какви са предимствата на абсолютния обектен код (поне 2)?

421. 1) {103} Предаване на адреса и данните
421. 2) {121} – По-бързо изпълнение на програмата.
– Налага се Компилиране само веднъж.
421. 3) {123} По-бърз за изпълнение с по-малък процент за грешка

422. 570. Какъв е основният проблем на абсолютния обектен код?

422. 1) {104} Основният проблем на абсолютния обектен код че редактирането и промяна е много трудна до невъзможна
422. 2) {104} Абсолютния обектен код може да работи само в една посока на действие.
422. 3) {121} Проблемът при него е че той служи само за заглавия е неможем да го използваме за друго
422. 4) {121} Абсолютния обектен код участва във всички аритметични операции.

423. 571. Кой вид адресация е източник на проблемите на абсолютния обектен код?

423. 1) {113} косвено абсолютно адресиране
423. 2) {121} Адресиране по база.
423. 3) {122} Относителната адресация.
423. 4) {125} Привързаността на МП към определени адреси от ОП

424. 575. По какво се различават абсолютните от преместваемите имена в езика Асемблер?

424. 1) {114} Различават се по начина на работа.
424. 2) {121} Абсолютните не могат да бъдат преместени, докато при преместваемите е обратно.
424. 3) {122} Абсолютните имена могат да участват във всички аритметични операции а преместваеми имена могат да участват само в операции събиране и изваждане
424. 4) {124} Абсолютните имена не се променят

425. 576. В какви операции могат да участват абсолютните имена? Защо?

425. 1) {121} Във всички видове адресиране, защото при абсолютен обектен код ясно се задава адресът на зареждане на програмата в паметта.

426. 577. В какви операции могат да участват преместваемите имена? Защо?

426. 1) {121} Преместваемите имена могат да участват в операции като: копиране, изрязване и т. н.Преместването на име позволява на друг файл да има същото име.
426. 2) {124} Във всички аритметични, както и логически.
426. 3) {124} Те могат да участват само в събиране, изваждане, конюнкция **и др**.

427. 578A. Какви са предимствата на преместваемия обектен код (поне 2)?

427. 1) {104} Многократно решаване
427. 2) {111} 1) Не е обвързан с конкретен ОП код.
2) Не е необходим превод при пресмятане
427. 3) {111} Обем на програмата значително намалява
427. 4) {121} Предимствата на преместваемия обектен код са че той е
427. 5) {122} МП е напълно готова за изпълнение въвеждането на МП в ОП е по-бързо
427. 6) {125} – адресите не са фиксирани
– ОП може да се ползва изцяло, а докато при абсолютният обектен код, който се записва в ОП не можеше.

428. 578В. Какви са недостатъците на преместваемия обектен код (поне 2)?

428. 1) {111} а) бавни са в) . . .

428. 2) {113} ① трудности при писане на код.
② води до логически грешки

428. 3) {121} Програмата се изпълнява по-бавно.

428. 4) {122} Проблема при тях е, че:
Води до бъгове при честото му преместване
Заема допълнително памет

429. 579. Какви изменения настъпват в обектен код „образ на паметта“ за да стане той преместваем?

429. 1) {104} Трябва да бъдат променени адресите на клетките.

430. 580. Как се въвежда в ОП преместваем обектен код „образ на паметта“?

430. 1) {103} Получава се адрес на зареждане, текстът на програмата се прочита в ОП от този начален адрес. Четат се броят и списъка на корекциите

431. 584. Как се въвежда в ОП преместваем обектен код „поредица от записи“?

431. 1) {103} МП се разделя на няколко текстови записа, за да се преодолее ограничението в размера на физическия запис във ВП. Това се нарича поредица от записи
– получава се адрес на зареждане
– отново се пропускат записите до намиране
– на заглавен запис с име на желаната програма

НЕПЪЛНИ И ЧАСТИЧНО СМЕШНИ ОТГОВОРИ

432. 001. Какъв е приносът на Чарлз Бебидж към компютърните системи?

432. 1) {106} Бабидж създава диферентна машина през 1822. По-късно обмисля създаването на аналитична, **която успява да реализира**.

433. 003В. Защо Огъста Едъ се смята за първия компютърен програмист?

433. 1) {110} Защото е написала първата ковпютърна програма.

433. 2) {112} Защото измисля програма за изчисляване на аритметични задачи

434. 007. Какви са трите принципа на Джон фон Нойман?

434. 1) {102} 1)2-инч бр. с-ма 2) програмата да се съхранява 3) изцяло изчисления

434. 2) {113} Двоична бройна система II Програмата се съхранява в паметта III достатъчна е само операция събиране

435. 008. Кои два въпроса трябва да получат принципен отговор за да бъдат реализирани идеите на Ч. Бебидж за създаване на компютър – автоматично работеща сметачна машина?

435. 1) {110} – Как ще бъдат представени числата?
– Кой ще ги управлява?

436. 021. Как по друг начин бихте нарекли компютрите със специално предназначение?

436. 1) {114} Специални

437. 026. По какъв критерий компютрите се делят на поколения?

437. 1) {102} Спрямо технологията на изработка (релета, лампи, интегрални схеми), спрямо бързодействието на процесора и паметта и спрямо броя на процесорите

437. 2) {102} Релета, ел. лампи, транзистори, ис …

438. 048. Какво е предназначението на ОП?

438. 1) {103} Съхраняване на получените резултати от програмата
438. 2) {105} Оперативната памет служи за записване на програмата управляваща компютърната система, за запис на параметри и резултати.

439. 053. Абсолютно еднакви ли са всички клетки на ОП? Ако не – защо, ако да – как се различават?

439. 1) {124} Да ,всички клетки в ОП са абсолютно еднакви **[Как тогава се различават помежду си?]**
439. 2) {124} Не **[Защо?]**
439. 3) {126} Да, различават се чрез

440. 054. От какво се ръководят конструкторите, когато определят размера на клетката в ОП на даден компютър?

440. 1) {106} Конструкторите се ръководят **от типа** на най-често използваните данни.
440. 2) {124} Всяка клетка може да има различен брой битове, следователно размерът на клетката се определя от типа на стойността ѝ.

441. 056. Какъв е размерът на клетките на съвременните ОП и как се наричат те?

441. 1) {102} В съвремените размерите на клетките са 32 bit, 64 bit, 84 bit и 128 b;t което дава и съответното наименование
441. 2) {106} Размерът на клетката е n = 8 и се нарича **бит**.
441. 3) {125} Размерът на ОП в днешно време се измерва в гигабайт-ове. И се наричат (RAM) Оп с висока степен на интеграция
441. 4) {126} Размерът на клетките е n=8 и се наричат **битове**.
441. 5) {126} Размера е 8, нарича се бит

442. 057. Какви са характерните черти (поне 3) на ОП?

442. 1) {124} Характерните черти на ОП:
– Времето за достъп до произволна клетка
– Висока скорост на обмен на данни.
– Малък обем
– Висока цена (преодолява се)

443. 061. Какъв вид памет е ОП на съвременните компютри: адресна или асоциативна? Защо?

443. 1) {102} Адресна, защото е по-бърза, по-евтина, по надеждна е
443. 2) {103} Адресна защото така всеки файл си има адрес чрез който може да бъде намерен и използван.
443. 3) {105} Адресна. Защото заедно с данните се предават и адресите.
443. 4) {105} Използва се адресна памет. Тя се изполва и по-широко. Бърза, евтина и надежна.
443. 5) {105} Адресна всяка клетка от паметта има точно определен адрес, най-често започващ от 0.
443. 6) {113} По-удобно е да бъде асоциативна
443. 7) {114} Асоциативна
443. 8) {121} Адресна, за да може еднозначни да се идентифицира всяка клетка и да не е нужно при четене да се задават част от данните.
443. 9) {121} Адресна. Защото е по-бърза,
443. 10) {121} По-добре асоциативна
443. 11) {123} Паметта на съвременните компютри е асоциативна, защото се използва асоциативен принцип за посочване на място в паметта.
443. 12) {124} Адресна **[Защо?]**
443. 13) {124} Адресна, защото е съвременна.

443. 14) {126} адресна – по удобен достъп и голям обем

444. 065. Какви технологии могат да се използват за направа на ОП?

444. 1) {102} Релета и лампи, интегрални схеми
444. 2) {106} Електрическа, транзисторна и кондензаторна

445. 066. Какви са предимствата и недостатъците на електрическите памети?

445. 1) {114} Предимства: много бързи, компактни, лесни за изработки.

446. 067. Какво е следствието от използване на електрически принципи за запомняне при паметите?

446. 1) {101} Имат нужда от ел. ток, изключването анулира съдържанието на ОП. При включване в ОП няма програма
446. 2) {105} При електрическите памети имаме енергозависимо помнене. Тоест при спиране на електрическия поток ОП се нулира и в нея вече няма програма. Ел. памети са много бързи.

447. 071A. Посочете поне две предимства на схемите динамична памет в сравнение със схемите статична памет.

447. 1) {102} С динамичната памет се работи по-бързо и използва се и днес
447. 2) {109} Възможност за добавяне и промяна на динамичната памет
447. 3) {118} Съхранява адреси от данни
447. 4) {123} Могат да **обработват** различни данни, по-бързи са.

448. 071C. Посочете поне две предимства на схемите статична памет в сравнение със схемите динамична памет.

448. 1) {104} по-бърза, по достъпна
448. 2) {115} – консумира по-малко енергия;
– по-евтина.

449. 072. Кои видове интегрални схеми изменяема памет – статична или динамична, са предпочитани в съвременните компютри и защо?

449. 1) {104} Динамична защото информацията в интегралните схеми могат да бъдат премествани и пренареждани.
449. 2) {123} Динамична, защото позволява изменение на информацията, зададена в нея.

450. 076. Каква е връзката между размера на думата и броя на битовете в клетките на един компютър? Защо?

450. 1) {124} Битовете определящи размера на думата са кратни на размера на клетка в ОП. **[Защо?]**

451. 091. Определете операция „допълнение до 1“.

451. 1) {102} Инвертиране на битовете и добавяне на 1 към резултата
451. 2) {103} Ако крайния резултат не е удовлетворяващ задействаме операция която запълва резултата до желаната стойност.

452. 124. Кой вид грешка е постоянна при представяне с фиксирана запетая? Защо?

452. 1) {122} Абсолютната .– защото **представянето на оста е полево**.

453. 127. Коя аритметична операция е „опасна“ при представяне на числа с плаваща запетая и защо?

453. 1) {107, 107} Събиране на близки по абсолютна стойност числа с различни знаци **[Защо е опасно?]**

454. 133. При наличие на двоичен суматор, необходимо ли е да се разработва отделен суматор за работа с ДКД числа? Защо?

454. 1) {102} Не е необходимо понеже числата са винаги кратни и всички действия могат да се извършат само с действие събиране.

455. 149. Как по цифров път се представя звук?

455. 1) {110} Като се измери амплитудата и се представи дискретен вид

456. 152. От какво се изгражда една машинна програма?

456. 1) {109} Изгражда се от МЕ и МИ. Като в МЕ са правилата с които компютърът изпълнява МИ.

457. 154. Какво представлява понятието „машинен език“?

457. 1) {114} Това е език, който централния процесор разбира. Съдържа 0 и 1.

458. 159А. Защо в едно АП е най-естествено да бъде записан пълен адрес от ОП?

458. 1) {106} защото така директно се посочва мястото, където е записана необходимата информация

458. 2) {114} Защото това е пълният адрес на който се паси в ДП съответният операнд, но той е много дълъг и често част от началните му цифри са „0“.

458. 3) {121} Важно е да се представи пълна **адрес за операцията**

458. 4) {124} Така е най-лесно за разбиране, но не винаги се прави

458. 5) {124} Най-естествено е да се записва пълен адрес в едно адресно поле, защото така най-бързо се открива дадената клетка.

459. 160. Какви видове МИ има? Защо?

459. 1) {110} И, ИЛИ, НЕ-И, НЕ-ИЛИ.

459. 2) {114} Машинните инструкции са два вида

1) Обработващи за извършване на определени пресмятания: събиране, изваждане и др.,

2) Управляващи за анализиране на възникналите обстоятелства и вземане на решение. Два са **защото това са двата компонента на МИ**

459. 3) {124} Чрез МИ могат да бъдат контролирани повечето операции:

459. 4) Има МИ за преход (условен или безусловен), извикване на подпрограма, извършване на математически операции и достъп до върха на системния стек и части от паметта.

Видовете МИ са нужни на програмиста, за да контролира По-добре изпълнението на програмата.

460. 162. Защо е полезно да бъде намален броят на АП в МИ?

460. 1) {102} За да се изпълни по-бързо МИ.

460. 2) {106} Спестява се място, по-бързо се обработва информацията

460. 3) {109} Полезно е да бъде намален броят на АП в МИ защото по този начин се намалява размера на информацията следователно се заема по-малко памет

460. 4) {119} За повишаване на бързодействието.

460. 5) {124} Защото ЦП чете МИ и колкото те са по-къси по-бързо ще е техното изпълнение

460. 6) {124} За да няма конфликти помежду им

461. 167. Как влияе елиминирането на третото АП (за резултат) на конструирането на ЦП?

461. 1) {115} Резултата **се пренася** по първия операнд

462. 170. Може ли една МИ да няма АП? Ако да – как, ако не – защо?

462. 1) {102} Да – ако се зарежда директно от процесора, като например BIOS

462. 2) {111} Да може, стига резултатът и операндите да се разграничават в КОП.

462. 3) {124} да

463. 173. Какво е предназначението на ЦП?

463. 1) {110} Да чете и изпълнява машинни инструкции от паметта.
463. 2) {121} Предназначението на централния процесор е да изпълнява машините програми чрез зададените от тях инструкции (алгоритми).
463. 3) {123} Централният процесор изпълнява зададената програма
463. 4) {123} Предназначението му е да изпълнява програмния код.

464. {123} Какво представлява понятието „микропроцесор“?

464. 1) {102} допълнителен процесор

465. 176. Какво е предназначението на УУ на ЦП?

465. 1) {111} УУ знае машинния език и **организира управление** на машинната програма

466. 183. Какво е предназначението на регистъра на инструкциите на ЦП?

466. 1) {106} Съхранява и **изпълнява**

467. 189. Посочете основните програмно достъпни регистри на ЦП.

467. 1) {102} **УУ**, **АЛУ**, ПБ.
467. 2) {115} Указанел на стек, ИР, **РАП**
467. 3) {126} Основни програмно достъпни регистри:
- Регистър условия (РУ)
- Регистър адреси от паметта (РАП)
- Регистър с общо предназначение (РОП)
- Регистър задачи (РЗ)

Регистрите са временна памет в ЦП

468. 196. От какви фази се състои изпълнението на една МИ?

468. 1) {124} От 3 – извличане, четене, изпълнение

469. 209В. Какво е следствието от различието на МЕ?

469. 1) {105} Различни програми, различни микропроцесорни системи, различен начин на адресация, различни инструкции.
469. 2) {112} **Микропрограмата** на един ЦП е абсолютно неразбираема за друг ЦП

470. 241. Какви проблеми създава поставянето на пълен адрес от ОП във всяко АП?

470. 1) {111} Поставянето на пълен адрес от ОП във всяко АП води до препълване на ОП, защото целият адрес е от 32 bit и забавя работата. Затруднява се обработването ва данни. в послед. адреси
470. 2) {125} Пълният адрес увеличава времето затърсене и използва повече ресурсна мощ.

471. 251. Възможно ли е един ЦП да предоставя само преки способи за адресиране? А полезно ли е?

471. 1) {123} Да възможно е

472. 284. Какви проблеми създава използването на косвена регистрова адресация при обхождане на вектор?

472. 1) {113} използването на 2 машинни инструкции също така и възможност за увеличаване

473. 304. Съществуват ли еквивалентни видове еднокомпонентни адресации? Ако не – защо, ако да – по-сочете ги и обяснете защо са еквивалентни.

473. 1) {121} Да

474. 334. Допуска ли относителното адресиране някакви модификации? Ако не – защо, ако да – кога и какви?

474. 1) {105} Да. При някои процесори.
474. 2) {106} Да, допускат се модификации **[Кога и какви?]**
474. 3) {121} Да, допуска **[Кога и какви?]**

475. 375. Какво представляват математическите съпроцесори (копроцесори)?

475. 1) {101} Ако се оперира УУ на TsP на кристала остава място за реализиране на АЛУ с ПЗ

476. 423. Само за предаване на параметри ли се използва апаратният стек? Ако да – защо, ако не – за какво друго се използва той?

476. 1) {123} Апаратният стек не се използва само за предаване на параметри

477. 437. Има ли ПУ, които могат да работят както като входни, така и като изходни? Ако не – защо, ако да – как се наричат тези устройства?

477. 1) {113} Да има, **принтери** и факсове
477. 2) {114} Да. Компактно комбинирани устройства. @D Pdm.
477. 3) {120} Има – наричат се комуникационни.

478. 439. Защо е необходима външна памет в компютрите?

478. 1) {110} а) Тя е енергонезависима
б) Има достатъчно голям капацитет

479. 441A. Посочете поне две предимства на магнитната в сравнение с оптическата технология за помнене.

479. 1) {102} Магнитната има по-голямо предимство за помнене, а оптичната има по.малко защото тя оптически помни много по-малко
479. 2) {110} Информацията може да се променя; Може да помни по-голям размер информация
479. 3) {110} По-дълъг живот, повече презаписвания

480. 443A. Посочете поне две предимства на твърдите в сравнение с гъвкавите магнитни дискове.

480. 1) {116} Енерго независими, повече повърхности (пакет дискове)

481. 472. Всички видове прекъсвания ли трябва да предоставя един ЦП? Защо?

481. 1) {114} не първите системи са предоставяли само прекъсвания за вход/изход
481. 2) {121} Не е нужно да представя всички видове прекъсвания. Трябва да представя само и единствено обикновеното прекъсване. Все пак най-важно е да се разбере, че има грешка.

482. 484B. Посочете поне две предимства на езика Асемблер в сравнение с МЕ.

482. 1) {126} По „четим“ за човек и по-кратък за изписване, затова и се създават по-бързо програми.

483. 485. Един единствен език Асемблер ли съществува? Ако да – защо, ако не – по какво съществено се различават отделните езици асемблер?

483. 1) {111} Не. Различават се в зависимост от процесора, на който ще се изпълнява.
483. 2) {111} Всеки ЦП има собствен език Асемблер, но **не се различават съществено** и могат да се изучават заедно.
483. 3) {121} Не, съществуват повече от един език Асемблер Те са сходни
483. 4) {121} Да.
483. 5) {121} Асемблер не е единствения език. Съществуват още езици и всички те се различават по начина на оформяне начин на кодиране, съхранение. Какви функции изпълнява, в каква сфера да се използват тези езици и т. н.
483. 6) {124} Не, няколко вида са и се различават според това, на какъв процесор ще функционират

484. 487A. Посочите поне две предимства на езика Асемблер в сравнение с език от високо равнище, който познавате добре (Паскал, Си, ???).

484. 1) {121} на Асемблер се пише по-бавно. **[Страхотно предимство!]** На ЕПВР се пише по-бързо. На Асемблер писането е затруднено. **[Още едно страхотно предимство!]**

484. 2) {124} Разбираема за хората

– Самата програма се пише по-лесно и по-бързо

485. 487B. Посочете поне два недостатъка на езика Асемблер в сравнение с език от високо равнище, който познавате добре (Паскал, Си, ???).

485. 1) {110} Не толкова лесно разбираем (четим) от хората като ЕПВР. При асемблер може да достигнем до всички МИ което понякога довежда до грешки.

485. 2) {110} По-бавно програмиране поради ниското равнище на езика за програмиране (Изисква повече време за написването на една програма.)

2) По-малко гъвкавост-

3) Заема повече памет от програма на език от по-високо равнище

486. 487C. Посочите поне две предимства на език от високо равнище, който познавате добре (Паскал, Си, ???) в сравнение с езика Асемблер.

486. 1) {121} Писането на код е по-лесно и по-гъвкаво. Има повече възможности. Компилацията и изпълнението на кода е по-стабилно

487. 487D. Посочите поне 2 недостатъка на език от високо равнище, който познавате добре (Паскал, Си, ???) в сравнение с езика Асемблер.

487. 1) {113} При Асемблер принос е, че на един ред от програмата отговаря една МИ. Програмата при ЕПВР е по-бавна и освен това изисква освен компилация и интерпретация

488. 491. Какво може да бъде записано в полето за мнемоничен код на операция при езика Асемблер?

488. 1) {114} Име на МИ

489. 499. Как в езика Асемблер може да бъде посочван размерът на обработваните данни?

489. 1) {103} Чрез използването на акумулатор

489. 2) {111} Записва се допълнителна буква към мнемоничния код на операция. Също така може да се посочва **явно или неявно от параметъра**.

490. 502. Как става определянето (дефинирането) на едно символично име в езика Асемблер?

490. 1) {103} символно име еквивалентно на адрес ‘а’ се дефинира като се запише в полето за етикет на този оператор

491. 536. Какви са входните данни на транслаторите от Асемблер и Макроасемблер?

491. 1) {126} Основен вход е текстът на програмата и евентуално библиотека с макродефиниции

492. 569B. Какви са предимствата на абсолютния обектен код (поне 2)?

492. 1) {126} Предимства: МП е напълно готова за изпълнение. Недостатъци: МП се привързва към конкретни адреси от ОП

СТАТИСТИЧЕСКИ ДАННИ

Общ брой на студентите, участващи в тази рубрика: 2080. **Нови** участници: 6.

Десетте върхови (**top 10**) постижения принадлежат на:

1. **126 точки: Станислав Георгиев Подскокниев=**, ф№0426068, [62+2] (1)
2. **114 точки: Мартин Живков Желев=,** ф№0601261086, [56+2] (от 2 място)
3. **100 точки: Светослав Иванов Чонков=,** ф№0526506, [50+0] (от 3 място)
4. **87 точки: Анна Василева Василева=**, ф№0318018, [43+1] (от 4 място)
5. **83 точки: Йордан Георгиев Панов=**, ф№0901262029, [41+1] (от 5–6 м.)
   83 точки: Владимир Красимиров Михайлов=, ф№0604731002, [41+1] (5–6)
7. **81 точки: Иван Пенчев Влашки=**, ф№0526516, [40+1] (от 7–8 място)
   81 точки: Стефан Антонов Бофиров=, ф№0901261048, [40+1] (от 7–8)
9. **77 точки: Ивелина Тилева Иванова=**, ф№0801262047, [36+5] (от 9 м.)
10. **76 точки: Димитър Стефанов Димитров=**, ф№0701262020, [37+2] (от 10)

Водещите тройки в отделните възрастови категории са:

1. **Информатика, бакалавър–редовно: (126 т, 1 м.=) Станислав Георгиев Подскокниев**, ф.№0426068, **(114 т., 2 м.=) Мартин Живков Желев**, ф.№0601261086, **(100 т., 3 м.=) Светослав Иванов Чонков**, ф.№1001261068**.**
2. **Информатика**, **бакалавър–задочно: (83 т., 5–6 м.=) Йордан Георгиев Панов**, ф№0901262029, **(81 т., 7–8 м.=) Иван Пенчев Влашки**, ф№0526516**, (77 т., 9 м.=) Ивелина Тилева Иванова**, ф№0801262047**.**
3. **Математика и информатика**, **бакалавър–редовно: (87 т., 4 м.=) Анна Василева Василева**, ф№0318018, **(65 т., 15 м.=) Мария Вихренова Сарафова**, ф№0518026, **(58 т., 16 м.=) Ангелина Сашова Хаджиева**, ф№0901181004**.**
4. **Математика и информатика**, **бакалавър–задочно: (51 т., 28–30 м.=) Христина Романова Якова**, ф№0701182040**, (40 т., 60–64 м.=) Дянко Христов Султов**, ф№0418510**, (32 т., 103–113 м.⇓) Нели Паскалева Паскалева**, ф№0418525**.**
5. **Информатика**, **магистър–софтуерни технологии: (70 т., 12 м.=) Владимир Йорданов Джалов**, **(56 т., 20–21 м.⇓) Димитър Гавраилов Фалков**, ф.№0601417004**, (54 т., 23–26 м.⇓) Ивайло Тодоров Иванов.**
6. **Информатика**, **магистър–бизнес информатика с английски език: (36 т., 80–85 м.⇓) Диляна Атанасова Атанасова**, ф.№1001437008**, (32 т., 103–113 м.⇓) Карина Георгиева Капитанова**, ф.№1301437005, **(30 т., 118–131 м.⇓) Митко Любомиров Иванов**, ф.№1401437006**.**
7. **Математика и информатика**, **Смолян: (16 т., 391–468 м.⇓) Юлиана Момчилова**, ф.№063505**, Анелия Фиданова Мутева**, ф.№073512, **Зюмбюл Юсуфова Мутишева**, ф.№083505 и **Петър Николаев Томов**, ф.№083515, **(14 т., 486–585 м.⇓) Филип Николаев Караасенов**, ф.№ 043510**, Николай Иванов Калайджиев**, ф.№ 063517, **Боряна Атанасова Зидарова**, ф.№073509 и **Стефан Славов Кехайов**, ф.№103503**, (13 т., 585–606 м.⇓) Владимир Йорданов Станкев**, ф.№ 083507**.**
8. **Биоинформатика, Биологически ф-т ПУ: (83 т., 5–6 м.=) Владимир Красимиров Михайлов**, ф.№0604731002**, (72 т., 11 м.=) Желью Димчев Русев**, ф.№0704731019**, (40 т., 60–64 м.=) Георги Бориславов Димитров**, ф.№0704731013**.**

Най-добре представил се нов участник (без името му):

101 издание, 5 април 2013 г., 4 точки (2+0), 1204–1444 място от 1706;
102 издание, 20 юни 2013 г., 14 точки (7+0), 376–452 място от 1710;
103 издание, 27 юни 2013 г., 6 точки (3+0), 966–1176 място от 1713;
104 издание, 10 септември 2013 г., 26 точки (13+0), 128–145 място от 1719;
105 издание, 21 декември 2013 г., 25 точки (12+1), 146–155 място от 1759;
106 издание, 22 февруари 2014 г., 20 точки (10+0), 221–248 място от 1763.
107 издание, 20 март 2014 г., 12 точки (6+0), 507–601 място от 1775;
108 издание, 4 април 2014 г., няма нови участници;
109 издание, 26 април 2014 г., 15 точки (7+1), 392–405 място от 1780;
110 издание, 3 май 2014 г., 26 точки (13+0), 132–150 място от 1798;
111 издание, 23 юни 2014 г., 16 точки (8+0), 330–394 място от 1851;
112 издание, 29 юни 2014 г., 12 точки (6+0), 524–623 място от 1857 [2 броя];
113 издание, 30 юни 2014 г., 22 точки (11+0), 198–218 място от 1864;
114 издание, 16 септември 2014 г., 16 точки (8+0), 346–414 място от 1875;
115 издание, 14 декември 2014 г., 24 точки (12+0) [2 бр.], 175–194 м. от 1910;
116 издание, 21 декември 2014 г., 6 точки (3+0) [2 бр.], 1088–1326 м. от 1914;
117 издание, 23 март 2015 г., 6 точки (3+0), 1088–1327 м. от 1921.
118 издание, 16 май 2015 г., 8 точки (4+0) [4 бр.], 867–1040 м. от 1928;
119 издание, 31 май 2015 г., 18 точки (9+0), 284–334 м. от 1937;
120 издание, 18 юни 2015 г., няма нови участници;
121 издание, 22 юни 2015 г., 30 точки (15+0), 109–122 м. от 1982;
122 издание, 30 юни 2015 г., 14 точки (7+0), 459–553 м. от 1993;
123 издание, 10 септември 2015 г., 10 точки (5+0), 718–868 м. от 2001;
124 издание, 15 юни 2016 г., 20 точки (10+0), 718–868 м. от 2001 [2 бр.];
125 издание, 27 юни 2016 г., 16 точки (8+0), 387–464 м. от 2075;
126 издание, 10 септември 2016 г., 12 точки (6+0), 611–714 м. от 2080;

Брой на представените шедьоври:

64 броя – 1 (1) участник, с 8 участия, на 1 място в класацията.
58 броя – 1 (1) участник, с 10 участия, на 2 място в класацията.
50 броя – 1 (1) участник, с 5 участия, на 3 място в класацията.
44 броя – 1 (1) участник, с 5 участия, на 4 място в класацията.
42 броя – 2 (2) участника, на 5–6 място в класацията:
  с 4 участия – 1 (1) участник, на 5–6 място в класацията;
  с 6 участия – 1 (1) участник, на 5–6 място в класацията.
41 броя – 3 (3) участника, от 7–8 до 9 място в класацията.
  с 6 участия – 2 (2) участника, от 7–8 до 9 място в класацията;
  със 7 участия – 1 (1) участник, на 7–8 място в класацията.
39 броя – 1 (1) участник, с 5 участия, на 10 място в класацията.
36 броя – 3 (3) участника, от 11 до 13 място в класацията:
  с 3 участия – 1 (1) участник, на 12 място в класацията;
  с 6 участия – 1 (1) участник, на 11 място в класацията;
  с 8 участия – 1 (1) участник, на 13 място в класацията.
35 броя – 1 (1) участник, с 6 участия, на 14 място в класацията.
33 броя – 1 (1) участник, с 6 участия, на 15 място в класацията.
30 броя – 1 (0) участник, с 5 участия, на 19 място в класацията.
29 броя – 4 (4) участника, от 16 до 17–18 и на 22 място в класацията:
  с 3 участия – 1 (1) участник, на 16 място в класацията;
  с 4 участия – 3 (3) участника, на 17 –18 и 22 място в класацията.
28 броя – 2 (3) участника на 20–21 място в класацията.
  с 3 участия – 1 (1) участник, на 20–21 място в класацията;

с 4 участия – 1 (2) участник, на 20–21 място в класацията.
27 броя – 4 (4) участника, на 23–26 място в класацията:
с 3 участия – 2 (2) участника, на 23–26 място в класацията;
с 5 участия – 2 (2) участника, на 23–26 място в класацията.
26 броя – 4 (4) участника, от 27 до 28–30 място в класацията:
с 3 участия – 2 (2) участника, от 27 до 28–30 място в класацията;
с 4 участия – 1 (1) участник, на 28–30 място в класацията;
с 6 участия – 1 (1) участник, на 28–30 място в класацията.
24 броя – 6 (6) участника, от 31–35 до 36 място в класацията.
с 3 участия – 2 (1) участника, от 31–35 до 36 място в класацията;
с 4 участия – 2 (2) участника, на 31–35 място в класацията;
с 5 участия – 2 (1) участник, на 31–35 място в класацията;
с 6 участия – 1 (1) участник, на 31–35 място в класацията.
23 броя – 4 (4) участника, на 37–40 място в класацията:
с 3 участия – 2 (2) участника, на 37–40 място в класацията;
с 5 участия – 1 (1) участник, на 37–40 място в класацията;
с 6 участия – 1 (1) участник, на 37–40 място в класацията.
22 броя – 8 (8) участника, от 41–45 до 46–48 място в класацията:
с 2 участия – 3 (3) участника, от 41–45 до 46–48 място в класацията;
с 3 участия – 1 (1) участник, на 41–45 място в класацията;
с 4 участия – 2 (2) участника, на 41–45 място в класацията;
с 5 участия – 2 (2) участника, от 41–45 до 46–48 място в класацията.
21 броя – 12 (12) участника, от 49–56 до 57–59 и на 68 място в класацията:
с 2 участия – 2 (2) участника, на 49–56 и на 68 място в класацията;
с 3 участия – 5 (5) участника, от 49–56 до 57–59 място в класацията;
с 4 участия – 2 (2) участника, от 49–56 до 57–59 място в класацията;
с 5 участия – 2 (2) участника, на 49–56 място в класацията.
със 7 участия – 1 (1) участник, на 57–59 място в класацията;
20 броя – 11 (9) участника, от 60–64 до 65–69 и на 74–76 място в класацията:
с 2 участия – 2 (2) участника, от 60–64 до 65–68 място в класацията;
с 3 участия – 5 (4) участника, на 60–64 и на 74–76 място в класацията;
с 4 участия – 2 (2) участника, на 60–64 място в класацията;
с 5 участия – 1 (1) участник, на 65–67 място в класацията.
19 броя – 10 (11) участника, на 69–73, 77–79 и 86–87 място в класацията:
с 3 участия – 6 (7) участника, на 69–73, 77–79 и 86–87 място в класацията;
с 4 участия – 2 (2) участника, на 69–73 и 77–79 място в класацията;
с 5 участия – 1 (1) участник, на 69–73 място в класацията;
с 6 участия – 1 (1) участник, на 69–73 място в класацията.
18 броя – 8 (7) участника, на 80–85, 88 и 97 място в класацията:
с 2 участия – 2 (2) участника, на 80–85 и на 97 място в класацията;
с 3 участия – 2 (2) участника, на 80–85 място в класацията;
с 4 участия – 1 (0) участник, на 80–85 място в класацията;
с 5 участия – 3 (3) участника, на 80–85 и на 88 място в класацията.
17 броя – 13 (13) участника, на 89–96, 98–102 и 114 място в класацията:
с 1 участие – 2 (2) участника, на 89–96 място в класацията;
с 2 участия – 4 (6) участника, на 89–96, 98–102 и 114 място в класацията;
с 3 участия – 1 (1) участник, на 89–96 място в класацията;
с 5 участия – 2 (2) участника, на 89–96 и 98–102 място в класацията;
с 6 участия – 1 (1) участник, на 98–102 място в класацията;
с 8 участия – 1 (1) участник, на 98–102 място в класацията.
16 броя – 16 (15) участника, на 103–113, 115–117 и 132–133 м. в класацията:
с 2 участия – 7 (6) участника, на 103–113, 115–117 и 132–133 място в класацията;
с 3 участия – 5 (6) участника, на 103–113 и 115–117 място в класацията;
с 4 участия – 2 (2) участника, на 103–113 място в класацията;
с 5 участия – 1 (1) участник, на 115–117 място в класацията.
15 броя – 23 (23) участника, на 118–131, 134–140, 152 и 159 място:

с 1 участие – 1 (1) участник, на 118–131 място в класацията;
с 2 участия – 14 (15) участника, на 118–131, 134–140 и 152 място в класацията;
с 3 участия – 3 (4) участника, на 118–131 и 134–140 място в класацията;
с 4 участия – 3 (3) участника, на 118–131 и 134–140 място в класацията.
с 5 участия – 1 (1) участник, на 159 място в класацията.

14 броя – 17 (17) участника, на 141–151 и 153–158 място:
с 1 участие – 1 (1) участник, на 141–151 място в класацията;
с 2 участия – 5 (8) участника, на 141–151 и 153–158 място в класацията;
с 3 участия – 4 (5) участника, на 141–151 и 153–158 място в класацията;
с 4 участия – 2 (2) участника, на 141–151 и 153–158 място в класацията;
с 5 участия – 1 (0) участника, на 153–158 място в класацията;
с 6 участия – 1 (1) участник, на 153–158 място в класацията;

13 броя – 36 (36) участника, от 160–181 до 182–193 и на 218–219 място:
с 1 участие – 8 (10) участника, на 160–181 и 182–193 място в класацията;
с 2 участия – 15 (15) участника, на 160–181, 182–193 и 218–219 място;
с 3 участия – 10 (9) участника, на 190–181 и 182–193 място в класацията;
с 4 участия – 3 (2) участника, на 160–181 и 218–219 място в класацията;
с 5 участия – 0 (1) участника.

12 броя – 36 (35) участника, на 194–217, 220–228 и 250–252 м. в класацията:
с 1 участие – 6 (6) участника, на 194–217 и 250–252 място в класацията;
с 2 участия – 18 (17) участника, на 194–217, 220–228 и 250–252 място в класацията;
с 3 участия – 11 (11) участника, на 194–217 и 220–228 място в класацията;
с 4 участия – 1 (1) участник, на 194–217 място в класацията.

11 броя – 33 (28) участника, на 229–249, 253–260, 300–302 и 315 място:
с 1 участие – 7 (6) участника, на 229–249, 253–260 и 300–302 място в класацията;
с 2 участия – 13 (11) участника, на 229–249, 253–260 и 300–302 място в класацията;
с 3 участия – 9 (8) участника, на 229–249 и 253–260 място в класацията;
с 4 участия – 3 (3) участника, на 229–249, 253–260 и 315 място в класацията;
с 6 участия – 1 (1) участник, на 229–249 място в класацията.

10 броя – 52 (54) участника, на 261–299, 303–314 и 374 място в класацията:
с 1 участие – 19 (19) участника, на 261–299 и 303–314 място в класацията;
с 2 участия – 21 (23) участника, на 261–299, 303–314 и 374 място в класацията;
с 3 участия – 10 (9) участника, на 261–299 и 303–311 място в класацията;
с 4 участия – 2 (3) участника, на 261–299 място в класацията.

9 броя – 77 (75) участника, на 316–373, 375–390 и на 484–485 място:
с 1 участие – 27 (27) участника, на 316–373 и 375–390 място в класацията;
с 2 участия – 38 (38) участника, на 316–373, 375–390 и на 484–485 място;
с 3 участия – 9 (8) участника, на 316–373, 316–373 и на 375–390 място в класацията;
с 4 участия – 2 (2) участника, на 316–373 място в класацията.

8 броя – 96 (96) участника, от 391–468 до 469–483 и на 586–588 място:
с 1 участие – 46 (47) участника, от 391–468 до 469–483 и на 586–588 място;
с 2 участия – 36 (38) участника, на 391–468 и на 586–588 място в класацията;
с 3 участия – 10 (10) участника, от 391–468 до 469–483 и на 586–588 място;
с 4 участия – 1 (1) участника, на 391–468 място в класацията.

7 броя – 126 (126) участника, на 486–585, 590–610 и 715–718 място:
с 1 участие – 74 (75) участника, на 486–585, 590–610 и 715–718 място в класацията;
с 2 участия – 42 (41) участника, на 486–585, 590–610 и 715–718 място в класацията;
с 3 участия – 10 (10) участника, на 486–585, 590–610 и 715–718 място в класацията.

6 броя – 147 (144) участника, на 611–714, 719–753, 910–915 и 949 място:
с 1 участие – 93 (91) участника, на 611–714, 719–753, 910–915 и 949 м. в класацията;
с 2 участия – 46 (46) участника, на 611–714 и 719–753 място в класацията;
с 3 участия – 7 (7) участника, на 611–714 и 719–753 място в класацията.

5 броя – 201 (202) участника, на 754–909, 916–948, 1136–1146 и 1188 място:
с 1 участие – 156 (157) участника, на 754–909, 916–948, 1136–1146 и 1188 място;
с 2 участия – 42 (42) участника, на 754–909 и 916–948 място в класацията;
с 3 участия – 3 (3) участника, на 754–909 място в класацията;

4 броя – 238 (237) участника, на 950–1135, 1147–1187 и 1453–1463 място:
  с 1 участие – 193 (192) участника, на 950–1135, 1147–1187 и 1453–1463 място;
  с 2 участия – 43 (43) участника, на 950–1135 и 1147–1187 място в класацията;
  с 3 участия – 2 (2) участника, на 950–1135 място в класацията.
3 броя – 300 (299) участника, на 1189–1452, 1464–1494 и 1785–1789 място:
  с 1 участие – 258 (258) участника, на 1189–1452, 1464–1494 и 1785–1789 място;
  с 2 участия – 38 (37) участника, на 1189–1452, 1464–1494 и 1785–1789 място;
  с 3 участия – 4 (4) участника, на 1189–1452 място в класацията.
2 броя – 323 (323) участника, на 1495–1784, 1790–1821 и 2072 място:
  с 1 участие – 317 (317) участника, на 1495–1784, 1790–1821 и 2072 място;
  с 2 участия – 6 (6) участника, на 1495–1784 място в класацията.
1 брой – 258 (260) участника, на 1822–2071 и 2073–2080 място в класацията.