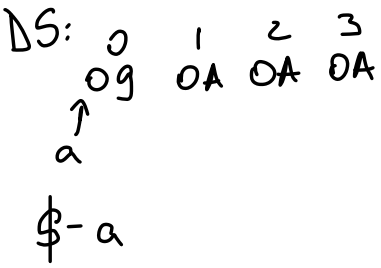
Ce valoare va avea c in urmatorul segment de date?
Which is the value of c in the following data segment?
segment data use32 class=data

a db 9
b times 3 db 10
c equ $-a

Select one:

- a. 5
- b. 3
- c. 1
- d. 4 ✔
- e. eroare sintactica
  syntax error

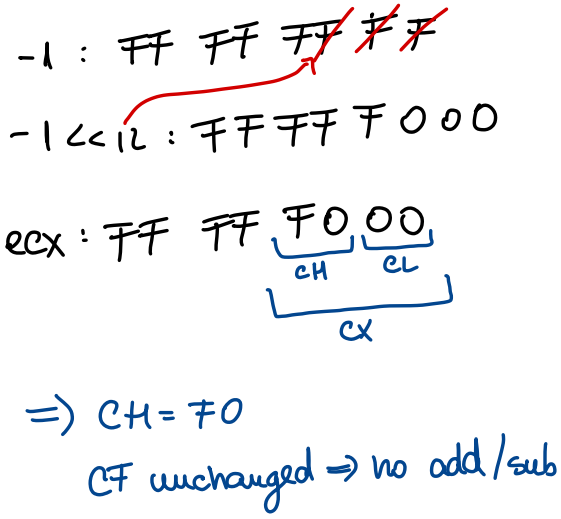


The correct answer is: 4

Ce valoare contine registrul CH dupa executarea instructiunii? Flag-ul CF este setat?
Which value is in CH after running the instruction? But in CF?
mov ecx, -1 << 12;?

Select one:

- a. ch = 0F0h; CF=1; ✖
- b. Syntax Error;
- c. ch = 0FFh; CF=nemodificat/unchanged;
- d. ch = 0F0h; CF=nemodificat/unchanged ✓
- e. ch = 0F0h; CF=0;

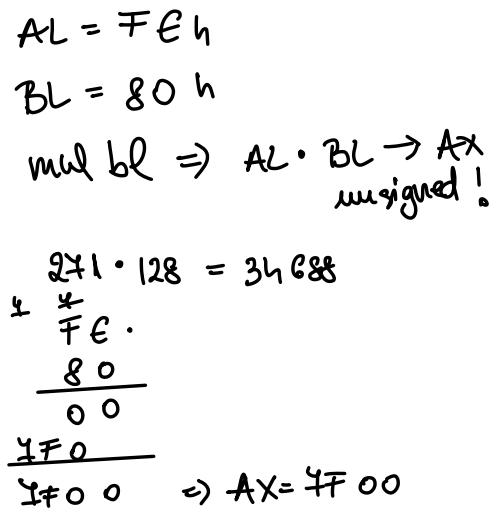


The correct answer is: ch = 0F0h; CF=nemodificat/unchanged

mov al, -2
mov bl, -128
mul bl
Rezultatul este:
The result is:

Select one:

- a. ax=100b
- b. ax=100h
- c. Assembly error
- d. Execution error ✖
- e. ax=7F00h ✓
- f. ax=FFFFh

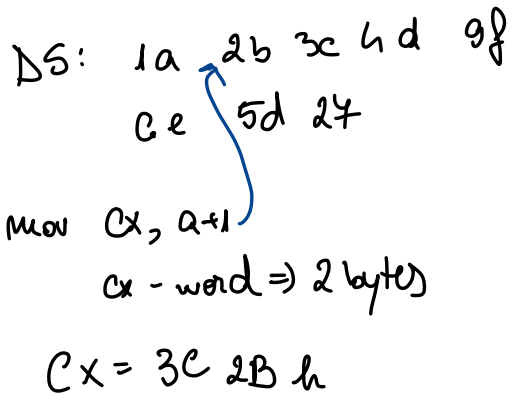


The correct answer is: ax=7F00h

Se da urmatorul segment de date:
The following data segment is given:
a db 1ah, 2bh, 3ch, 4dh, 9fh, 6eh, 5dh, 27h
Ce valoare va contine registrul CX in urma instructiunii:
What will be the value of CX after the execution of the instruction:
mov cx, [a+1]

Select one:

- a. eroare de sintaxa / syntax error
- b. 3c2bh ✔
- c. 4d3ch
- d. 2b3ch
- e. 3ch

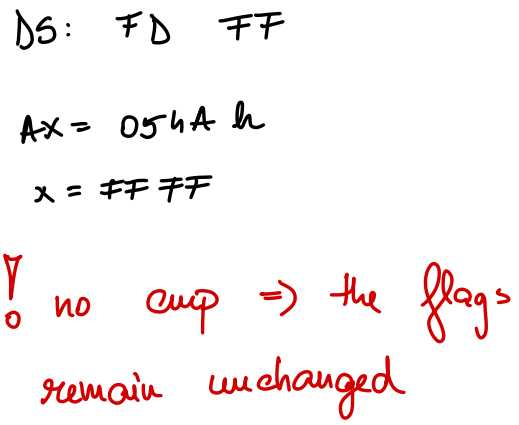


The correct answer is: 3c2bh

Fie urmatoarea secventa de cod
Consider the following code sequence
x dw 0fffdh
.....
mov ax,054ah
add byte [x], 2
jz a2
...
a2:...
Programul va
The program will

Select one:

- a. semnala eroare de sintaxa
  issue a syntax error
- b. executa un salt la adresa determinata de a2
  execute a jump to the address determined by a2
- c. executa un salt la adresa determinata de a2 numai daca distanta pana la eticheta destinatie nu depaseste 127 octeti
  execute a jump to the address determined by a2 only if the distance to the destination label is no more than 127 bytes
- d. nu va executa un salt la adresa determinata de a2 ✔
  not execute a jump to the address determined by a2
- e. semnala eroare de executie de tip "memory access violation"
  issue a "memory access violation" run time error



The correct answer is: nu va executa un salt la adresa determinata de a2
not execute a jump to the address determined by a2

Urmatoarea secventa este
The following sequence is
segment data
a dw 1,2,3,4,56,0ffeh
la equ ($-a)/2
b dw times 10 ($-b) syntax error

Select one:

- a. eroare logica
  logical error
- b. corecta
  correct
- c. eroare de sintaxa ✔
  syntax error
- d. eroare de alocare
  allocation error
- e. run time error

The correct answer is: eroare de sintaxa
syntax error

DS: 01 00 02 00 03 00
04 00 38 00 FE 0F

CF → se setează când operația depășește spațiul alocat

OF → 0+0=1 1+1=0
0-1=1 1-0=0

ADD/SUB → CF - unsigned OF - signed

MUL → OF (never), dar
OF = CF = 0 size(res) = size(op)
OF = CF = 1 altfel

DIV → OF = FATAL
OF / CF undefined

OF=1, SF=1 ⇒ CF=0

In urma executiei instructiunii add op1,op2 care din urmatoarele conditionari este adevarata ?
Which of the following conditionings is true after running the instruction add op1,op2 ?

Select one:

- a. If OF = 1 then (SF = 1 and CF = 0) ?
- b. If OF = 1 and SF = 1 then CF = 0 ✓
- c. nu exista nici o conditionare intre valorile celor trei flaguri
  there is no conditioning between the values of the three flags
- d. If OF = 0 and SF = 0 then CF = 0 ✖
- e. If OF = 0 and SF = 0 then CF = 1

OF=0 and SF=0 :
255+4 yet CF=1
-127+4 yet CF=0

OF=1 and SF=1 :

The correct answer is: If OF = 1 and SF = 1 then CF = 0

Care dintre urmatoarele instructiuni foloseste simultan atat adresarea directa la memorie cat si cea indirecta ? → you cannot ?
Which of the following instructions uses direct addressing and indirect addressing simultaneously ?

Select one:

- a. mov a,[ebx]
- b. mov ax,[ebx]
- c. mov [eax],bx
- d. nici una ✔
  none
- e. mov [a],ebx

The correct answer is: nici una
none

Urmatoarea instructiune:
The following instruction:

mov a, [eax]

Select one:

- a. incarca in a offset-ul operandului de memorie de la adresa gasita in EAX
  loads into a the offset of the memory operand from the address found in EAX
- b. incarca in a adresa NEAR desemnata de expresia [eax]
  loads into a the NEAR address designated by the expression [eax]
- c. este echivalenta cu lea a, [eax] ✗
  is equivalent to lea a,[eax]
- d. incarca in a offsetul de la adresa [eax] numai daca a este definita ca dublucuvant, in caz contrar fiind semnalata eroare de sintaxa
  loads into a the offset from [eax] only if a is defined as a doubleword, if not, a syntax error being issued
- e. nici una dintre variantele enumerate ✓
  none of the enumerated variants

The correct answer is: nici una dintre variantele enumerate
none of the enumerated variants

→ invalid combination of operands

MOV dest, source
↳ dest & source are either registers, variables of type byte, word or dword constants
↳ dest cannot be a constant
↳ both op. need to be of the same type
↳ at least one operand has to be a variable

Directivele indica:
The directives indicate:

Select one:

- a. modul in care sunt generate codul si datele in momentul executiei
  the way the code and the data are generated at execution time
- b. niciuna dintre cele patru variante specificate
  none of the four specified variants
- c. modul in care sunt generate codul si datele in momentul asamblarii ✓
  the way the code and the data are generated at assembly time
- d. doar modul in care sunt generate datele in momentul asamblarii
  just the way the data is generated at assembly time
- e. doar modul in care este generat codul in momentul asamblarii
  just the way the code is generated at assembly time

The correct answer is: modul in care sunt generate codul si datele in momentul asamblarii
the way the code and the data are generated at assembly time

→ Ex: DB, DW, DD, EXTERN, GLOBAL,

directives are indications to the assembler regarding How to process the source code and data

instructions are indications directed to the processor

Continutul registrului EFLAGS poate fi transferat in registrul EDX astfel:
The contents of the EFLAGS register can be transferred in the EDX register as follows:

Select one:

- a. pushf pop edx ✓
- b. push eflags ; pop edx
- c. nici un raspuns nu este correct ✗
  none of the specified answers is correct
- d. mov edx, eflags
- e. mov edx, [eflags]

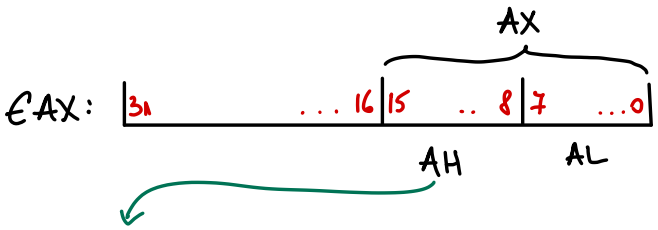
instruction to push the contents of EFLAGS onto the stack. the oposite is POPF

The correct answer is: pushf ; pop edx

Considerand bitul 0 cel mai putin semnificativ bit, prin executia carei instructiuni de mai jos bitii 9-12 initiali ai lui EAX vor fi identici cu bitii 0-3 ai lui AH de dupa executia instructiunii?
Considering bit 0 to be the least significant bit, by running which of the below instruction the bits 9-12 of the initial EAX will be identical with the bits 0-3 of the resulting AH?

Select one:

- a. Prin executia oricareia
  By running any one of them
- b. shr eax,1; ✓
- c. Nici una dintre instructiuni nu va produce efectul descris ✗
  None of the specified instructions will output the required result
- d. shl eax,1;



AH bits: 15 14 13 12 11 10 9 8 → one bit

AH: 7 6 5 4 3 2 1 0

⇒ shr eax, 1

The correct answer is: shr eax,1;

Rezultatul secventei de cod de mai jos e AX=-6?
The result of the following code is AX=-6 ?

```
mov al, -2
cbw
mov ch, 3
imul ch
```

Select one:

- a. Da ✓
  Yes
- b. Nu ✗
  No

AL = FE

cbw → AX = FFFE

CH = 03

FE · 03 = FA

FFFA = -6

3·14 = 42 | 16, 32 | 2 = A

2+F·3 = 3·15+2 = 47 | 16, 32 | 2 = F

The correct answer is: Da
Yes

Ce octeti se genereaza in memorie corespunzator urmatoarei declaratii?
Which are the bytes generated in memory for the following declaration?

```
a times 3 dw 10
b dd 0xAB
```

Select one:

- a. niciuna dintre variantele date
  none of the given variants
- b. | 0xA | 00 | 0xAB | 00 | 00 | 00 |
- c. | 10 | 10 | 10 | 0xAB |
- d. | 00 | 00 | 00 | 0xAB | 00 | 0xA | 00 | 0xA | 00 | 0xA |
- e. | 0xA | 00 | 0xA | 00 | 0xA | 00 | 0xAB | 00 | 00 | 00 | ✓

NO → 0A
dw ⇒ 2 bytes
⇒ a: 0A 00 x3

dd ⇒ 4 bytes ⇒ b: AB 00 00 00

memory: 0A 00 0A00 0A00 AB 00 0000

The correct answer is: | 0xA | 00 | 0xA | 00 | 0xA | 00 | 0xAB | 00 | 00 | 00 |

Fie urmatoarea instructiune
Consider the following instruction

mov AX,0ff00h

urmata de secventa:
followed by the sequence:

add AX,8000h
sbb dx,dx
sub AX,8000h

Efectul secventei date asupra registrului DX este echivalent cu
The effect of the given sequence on the DX register is the same as

Select one:

- a. mov DX,0ff00h
- b. nici una dintre variantele enumerate
  none of the enumerated variants
- c. cwd ✓
- d. cbw
- e. mov dx, 0 ✗

The correct answer is: cwd

AX = FF 00
add AX, 8000h; AX = 7F 00
CF = 1
sbb dx, dx ; DX = FF FF
cwd:
AX = FF 00
takes the sign from AX (1) and replaces
DX with FF FF if SF = 1
or 00 00 if SF = 0

Operandul [ebx*3] reprezinta: The operand [ebx*3] represents:

Select one:

- a. ~~un~~ un operand specificat in mod adresare la memorie bazat - indexat cu factorul de scala 3
  a memory addressing operand based and scaled indexed by factor 3
- b. ~~un~~ un operand specificat in mod adresare directa la memorie, bazat-indexat cu factorul de scala 2
  a memory direct addressing operand based and scaled indexed by factor 2
- c. ~~un~~ un operand specificat in mod registru
  a register mode operand
- d. un operand specificat in mod adresare indirecta la memorie, bazat-indexat cu factorul de scala 2 ✓
  a memory indirect addressing operand based and scaled indexed by factor 2
- e. ~~un~~ un operand specificat in mod adresare la memorie indexat cu factorul de scala 3 ✗
  a memory addressing operand scaled indexed by factor 3

The correct answer is: un operand specificat in mod adresare indirecta la memorie, bazat-indexat cu factorul de scala 2
a memory indirect addressing operand based and scaled indexed by factor 2

[ebx * 3] ⟺ [ebx + ebx*2]
base, index, scale
⇒ base + index + scale·2
'indirect addressing!

mov al, -2
mov bl, -128
imul al
Rezultatul este:
The result is:

Select one:

- a. Execution error
- b. ax=FFFFh
- c. ax=FF00h ✗
- d. Assembly error
- e. ax=100h
- f. ax=100b ✓

The correct answer is: ax=100b

AL = FE
BL = 80
imul al ⇒ AL · AL
FE < 0 ⇒ neg AL = 02
02 ·
02
0004 ⇒ 100b.

De cate ori se repeta bucla again?
How many times the again loop will be repeated?
mov bx,0
mov ecx,0
again:
shl ecx,1
inc bx
loop again

goes into a ecx = 01 | 02 loop?

Select one:

- a. ciclu infinit ✔
  infinite loop
- b. 16
- c. 1
- d. 10
- e. 65535
- f. 0

The correct answer is: ciclu infinit
infinite loop

BX = 00 00
ECX = 00 00 00 00
shl ecx ⇒ ecx = 00000000
BX = 0001
ecx ← ecx − 1 = FF FF FF FF
shl ecx ⇒ ecx = FF FF FF FC
FC
F A
F 8
F 4
F 2
F 1
:
at every loop dec ecx ⇒ it never reaches 0

Ce valoare are contorul de locatii ($) la sfarsitul urmatoarelor declaratii de date (considerand ca offset-ul de inceput al segmentului de date este 0):
What will be the value of the location counter ($) at the end of the following data declaration (assuming that the starting offset of the segment is 0):
segment data ..
a times 3 db 2
l equ 3
b dw 10

Select one:

- a. 4
- b. 7
- c. 5 ✔
- d. 6

The correct answer is: 5

memory:

0	1	2	3	4	5
02	02	02	10	00	$

Care este efectul executiei instructiunii "mov [a], -1" in conditiile definitiilor:
Which is the effect of the execution of instruction "mov [a],-1" for the following data definitions:
segment data
a resw 1
b db 3Ch, 4Dh

Select one:

- a. a=0ff3Ch
- b. a=0ffffh
- c. a=00ffh
- d. eroare de sintaxa ✔
  syntax error
- e. a=3Cffh

The correct answer is: eroare de sintaxa
syntax error

mov byte [a], -1
mov word [a], -1
would have been correct

error: operation size not specified

Ce octeti se genereaza in memorie corespunzator urmatoarei declaratii (adresele de memorie cresc de la stanga la dreapta) ?
Which are the bytes generated in memory for the following declaration (memory addresses increase from left to right) ?

a times 2 dd 0xABCD

Select one:

- a. | 0xAB | 0xCD | 0xAB | 0xCD |
- b. | 0xCD | 0xAB | 00 | 00 | 0xCD | 0xAB | 00 | 00 | ✔
- c. | 00 | 00 | 0xAB | 0xCD | 00 | 00 | 0xAB | 0xCD |
- d. | 0xCD | 0xAB | 0xCD | 0xAB |
- e. niciuna dintre variantele date
  none of the given variants

memory:

CD AB 00 00 CD AB
00 00

The correct answer is: | 0xCD | 0xAB | 00 | 00 | 0xCD | 0xAB | 00 | 00 |

Instructiunea mov [a], word 2 cu a resb 1 isi exprima operanzii :
The instruction mov [a], word 2 with a resb 1 expresses its operands:

Select one:

- a. in mod adresare directa
  in direct addressing mode
- b. in mod adresare la memorie
  in memory addressing mode
- c. instructiunea specificata este incorecta sintactic ✖
  the above specified instruction is sintactically incorrect
- d. in mod imediat si adresare la memorie
  in immediate and memory addressing mode ✓
- e. in mod imediat si adresare indirecta
  in immediate and indirect mode

memory:

a resb 1
- - - - -

mov [a], word 2
↑ memory addressing mode
↓ immediate
(directly includes the data to be acted on)

The correct answer is: in mod imediat si adresare la memorie
in immediate and memory addressing mode

Calculul de adresa se efectueaza pe baza:
Address computation is done based on:

Select one:

- a. nici o varianta de calcul exprimata nu este corecta ✔
  none of the presented variants is correct
- b. valorii curente din registrii CS si EIP → instruction pointer
  the actual values from CS and EIP registers
- c. offseturilor asociate etichetelor de date, offset-uri deduse pe baza valorii curente a contorului de locatii
  the offsets associated to data labels, offsets inferred based on the actual location counter value
- d. valorii etichetelor de cod asociate instructiunilor de salt in cadrul segmentului curent
  the code labels values associated to jump instructions inside the current segment
- e. adresei de segmentare dedusa din valoarea selectorului de segment asociat
  the segmentation address inferred from the associated segment selector

The correct answer is: nici o varianta de calcul exprimata nu este corecta
none of the presented variants is correct

Ce va fi in DX in urma executiei urmatoarelor instructiuni:
What will be the value of DX at the end of the following code:
mov ax, 0ffffh
cwd
add dx, 1
mov bx, 65535
div bx

Select one:

- a. 65535
- b. 0 ✔
- c. 1
- d. ffffh
- e. 00ffh

The correct answer is: 0

AX = FF FF

DX:AX = FF FF FF FF

DX = 00 00

BX = FF FF

DX:AX / BX = 1

⇒ AX = 00 01

DX = 00 00

Care este valoarea din AH dupa executia instructiunii " mov ah, (2&7)^(23^(~31))":
Which is the value from AH after running the instruction " mov ah, (2&7)^(23^(~31))":

Select one:

- a. 05fh ✖
- b. 0
- c. 0ffh
- d. eroare de sintaxa
  syntax error
- e. 0f5h
- f. 1

The correct answer is: 0f5h

(2&7) ^ (23 ^ (~31))

31 → $2^5 - 1$ = 00011111b

~31 = 11100000b

23 = 00010111

23 ^ (~31): 00010111 ^
11100000
11110111

00000108
00000111
00000010

⇒ 00000010 ^
111 10111
111 10101 = F5h

Instructiunea mov eax, [ebx+esp] este
The instruction mov eax, [ebx+esp] is

Select one:

- a. eroare de sintaxa datorata faptului ca ESP este singurul registru general ce nu poate fi folosit in adresarea indirecta
  syntax error due to the reason that ESP is the only general register that can not be used in indirect addressing
- b. eroare logica
  logical error
- c. corecta atat sintactic cat si logic doar daca se precizeaza explicit registrul segment la care se raporteaza (DS sau SS)
  syntactically and logically correct only if there is an explicit specification of the corresponding segment register (DS or SS)
- d. corecta ✔
  correct
- e. eroare de sintaxa datorata ambiguitatii baza-index
  syntax error due to index-base ambiguity

The correct answer is: corecta
correct

[ebx + esp]
↑ ↑
index base

Since ESP cannot be an index ⇒ correct 2 AM formula

a dw 0ABCDh, 0FEDCh
---
mov esi, a
lodsb
lodsw
AX=?

Select one:
- a. 0DCABh ✔
- b. 0ABCDh
- c. 48DCh
- d. 48ABh

memory:
CD AB DC FE
esi esi esi
lodsb ⇒ AL = CD
lodsw ⇒ AX = DCAB

AX = DC ABh

The correct answer is: 0DCABh

---

mov ax, 65535
mov bl, 10
div bl
Rezultatul este:
The result is:

Select one:
- a. Execution error ✓
- b. ax=0000h
- c. ah=05h; al=1999h ✖
- d. ah=05h; al=00h
- e. Assembly error

AX = FF FF
BL = 0A
div BL
65535 / 10 = 6553
but on AL you can at most fit 255 unsigned
⇒ execution error

The correct answer is: Execution error

Indicatorii continuti in registrul EFLAGS sunt:
The contents of the EFLAGS register is represented by:

Select one:
- a. CF, DF, IF, SF, ZF, AF, PF, OF TF
- b. nici una din variante ✖
  none of the presented variants
- c. OF, DF, JF, TF, SF, ZF, AF, PF, CF, IF
- d. AF, CF, DF, IF, OF, EF, SF, TF, ZF
- e. CF, DF, PF, TF, ZF, SF, AF, IF, OF ✓

there are 9:
CF - carry flag
DF - direction flag
PF - parity flag
TF - trap flag
ZF - zero flag
SF - sign flag
AF - auxiliary flag
IF - interrupt flag
OF - overflow flag

The correct answer is: CF, DF, PF, TF, ZF, SF, AF, IF, OF

Se considera ca secventa de instructiuni se repeta de CX ori. Care secventa transfera valoarea din AX in BX?
We consider each set of instructions is executed CX times. Which sequence transfers the value from AX to BX?

Select one:
- a. shl ax,1; rcl bx,1; CX=15
- b. shl ax,1; rcl bx,1; CX=8 ✖
- c. shl ax,1; rcl bx,1; CX=7
- d. shl ax,1; rcl bx,1; CX=16 ✓

rcl = rotate with carry left
ex: 1101001010011101
shl, 1 ⇒ CF = most sign. bit
rcl: 1010010100111011
the size of AX is 16 bits
⇒ CX = 16

The correct answer is: shl ax,1; rcl bx,1; CX=16

De cate ori se repeta bucla again?
How many times the again loop will be repeated?
mov bx,0
mov ecx,0
again:
shr ecx,1
inc bx
loop again

Select one:

- a. 65535
- b. ciclu infinit ✖
  infinite loop
- c. 0
- d. 16
- e. 32 ✓
- f. 1

BX = 0
ECX = 0

shr: 0 111 .... .
it reaches 0 after 32 iterations and exits the loop

The correct answer is: 32

Ce valoare are contorul de locatii ($) la sfarsitul urmatoarelor declaratii de date:
Which is the value of the location counter ($) at the end of the following data declarations:
segment data
a db 2
b dw 3
c dd 10

Select one:

- a. 8
- b. 7 ✔
- c. 6
- d. 4
- e. 5

memory:

02	03	00	0A	00	00	00	$
0	1	2	3	4	5	6	7

The correct answer is: 7

Considerand bitul 0 cel mai putin semnificativ bit, izolarea bitilor 4-6 din registrul EAX se face folosind instructiunea?
Considering bit 0 to be the least significant bit, we can isolate bits 4-6 from EAX by using?

Select one:

- a. oricare dintre instructiunile specificate produc efectul dorit
  any of the mentioned instructions provide the aimed effect
- b. and EAX,112 ✔
- c. Nici una din instructiunile specificate nu produce efectul dorit
  None of the mentioned instructions provide the aimed effect
- d. xor EAX,112
- e. not EAX
- f. or EAX, 112

6 5 4 3 2 1 0
112d = 1110000b

⇒ and EAX, 112
would isolate bits 4-6

The correct answer is: and EAX,112

Directivele sunt indicatii date:
The directives are indications given to the:

Select one:

- a. asamblorului ✔
  assembler
- b. procesorului
  processor
- c. linkeditorului
  linkeditor
- d. depanatorului
  debugger
- e. sistemului de operare
  operating system

directives are indications to the [assembler] regarding how to process the source code and data

instructions are indications directed to the [procesor]

The correct answer is: asamblorului
assembler

Operanzii instructiunii de forma instr op1, op2:
The operands of the instruction instr op1, op2:

Select one:

- a. pot fi specificati unul in mod direct iar celalalt indirect → impossible
  may be specified one in direct mode and the other in indirect mode
- b. toate cele patru afirmatii sunt false ✔
  all the four answers are false
- c. pot fi amandoi specificati in mod indirect → mov [eax*2], [ebx+esp] not
  can be specified in indirect mode simultaneously
- d. nu pot fi specificati simultan in mod registru → add ax, bx
  may not be specified in register mode simultaneously
- e. nu pot avea dimensiuni diferite → not sure
  may not have different sizes

The correct answer is: toate cele patru afirmatii sunt false
all the four answers are false

mov ax, 65535
mov bl, 10
idiv bl
Rezultatul este:
The result is:

Select one:

- a. Assembly error
- b. Execution error ✖
- c. ah=FFh; al=00h ✓
- d. ah=05h; al=00h
- e. ah=05h; al=1999h

AX = FF FF
BL = 0A

AX signed int = -1

-1 | 10 ⇒ AL = 00
AH = -1

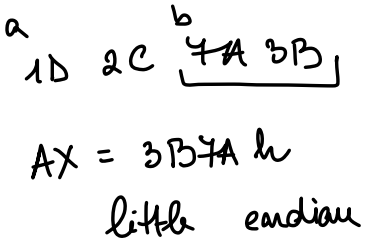
⇒ AH = FF
AL = 00

The correct answer is: ah=FFh; al=00h

Care este efectul executiei instructiunii "mov ax, [b]" pentru urmatorul segment de date:
Which is the effect of the execution of instruction "mov ax, [b]" for the following data segment:
segment data
a dw 2C1Dh
b db 7Ah, 3Bh

Select one:

- a. ax=7A3Bh ✖
- b. ax=1D2Ch
- c. ax=3B7Ah ✓
- d. ax=2C1Dh
- e. eroare de sintaxa
  syntax error



The correct answer is: ax=3B7Ah

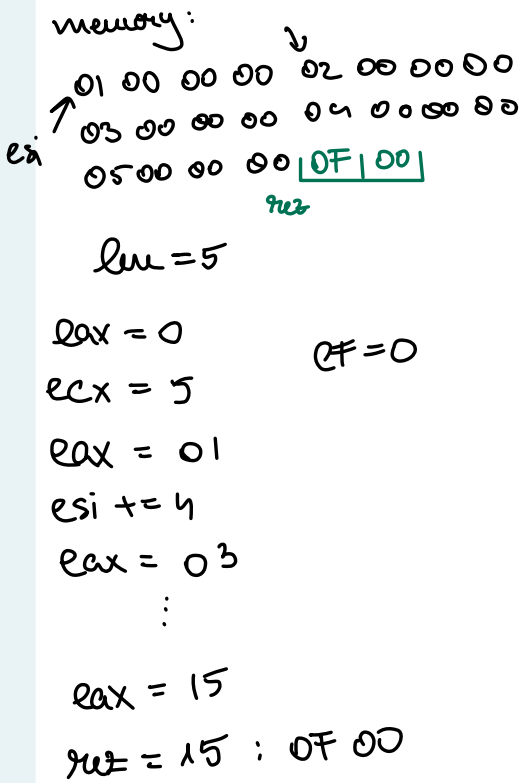
Urmatoarea secventa de cod va:
The following code sequence will:

segment data
sir dd 1,2,3,4,5
len equ 5
rez resw 1
...
segment code
...
lea esi, [sir]
mov eax, 0
mov ecx, len
clc

up:
adc eax, [esi]
inc esi
inc esi
inc esi
inc esi
dec ecx
jnz up
mov [rez], eax
...

Select one:

- a. secventa contine o eroare de sintaxa ✖
  the sequence will issue a syntax error
- b. determina suma elementelor din sir ✓
  compute sum of numbers from sir
- c. determina cel mai mare numar din secventa
  compute the largest number from the sequence
- d. determina cel mai mic numar din secventa
  compute the smallest number from the sequence
- e. determina diferenta elementelor din secventa
  compute the substraction of numbers from the sequence



The correct answer is: determina suma elementelor din sir
compute sum of numbers from sir

Segmentul de memorie:
The memory segment:

Select one:

- a. este o diviziune logica a memoriei unui program caracterizat prin adresa de baza, dimensiune si tip
  it is a logical section of a program's memory, featured by its basic address, size and type
- b. este un bloc fizic de memorie de lungime variabila caracterizat prin adresa de inceput si deplasament ✖
  it is a variable length physical memory block identified by its starting address and offset
- c. foloseste adrese pe 48 biti pentru reprezentarea adresei sale de inceput
  uses 48 bits addresses for representing its starting address
- d. are dimensiunea egala cu offset-ul sau
  has the size equal with its offset
- e. are dimensiunea maxima de 1 GB reprezentat pe adrese de 32 biti
  has the maximum size of 1 GB represented on 32 bits addresses

faculty's definition

A segment represents the *logical section* of a program's memory, featured by its *basic address*, by its *limit (size)* and by its *type*.

In 80x86-based processors, a segment can mean:

1. A block of memory of *discrete size* called a **physical segment** that has:
   - 64K bytes for 16-bits processors
   - 4G bytes for 32-bits processors
2. A variable-sized block of memory, called a **logical segment** occupied by a program's code or data

⇒ in this case it it a logical segment that has an address, size and type

The correct answer is: este o diviziune logica a memoriei unui program caracterizat prin adresa de baza, dimensiune si tip
it is a logical section of a program's memory, featured by its basic address, size and type

Care este efectul executiei secventei:
What is the execution effect of the following sequence:
mov al, 255
add al, -1

Select one:

- a. CF=1; OF=1;
- b. CF=0; OF=0;
- c. CF=0; OF=1;
- d. CF=1; OF=0; ✔

AL = FF

add AL, -1

AL + AL : FF + FF = 1FEh

510 | 16 → 496 | 31 | 16 → =14 | 16 | 1 | 16 → 15 | 1 | 0

510d = 1FEh

CF=1

but -+- = -
⇒ OF=0

The correct answer is: CF=1; OF=0;

Instructiunea
The instruction
sbb AL,AL
este echivalenta cu
is equivalent to

Select one:

- a. "mov AL,CF"
- b. shl AL,8
- c. "mov AL,ZF"
- d. nici una dintre variantele prezentate ✔
  none of the enumerated variants
- e. mov AL,0
- f. xor AL,AL
- g. "mov AL,DF"

"mov AL, CF"
—11—, DF doesn't exist
—11—, ZF

shl AL, 8
mov AL, 0
xor AL, 0
⇒ AL = 0

BUT we don't know the value of CF ⇒

sbb AL, AL = { FF, CF = 1 ; 00, CF = 0

The correct answer is: nici una dintre variantele prezentate
none of the enumerated variants

Codul sursa de mai jos calculeaza corect expresia (-1) * (-1) = 1
Does the next code compute correctly the expression (-1) * (-1) = 1
mov al, 0ffh
cbw
imul ax

Select one:

- a. Da ✔
  Yes
- b. Nu
  No

The correct answer is: Da
Yes

AL = FFh = (255 or -1)

cbw ⇒ AX = FF FF

imul AX ⇒ DX:AX = (-1)*(-1)
= 00 00 : 0001

Fie urmatoarea secventa de cod
Consider the following code sequence
x dw 0fffdh
.....
mov ax,054ah
add [x], 2
jz a2
...
a2:...
Programul va
The program will

Select one:

- a. executa un salt la adresa determinata de a2
  execute a jump to the address determined by a2
- b. semnala eroare de executie de tip "memory access violation"
  issue a "memory access violation" run time error
- c. semnala eroare de sintaxa ✓
  issue a syntax error
- d. nu va executa un salt la adresa determinata de a2 ✖
  not execute a jump to the address determined by a2
- e. executa un salt la adresa determinata de a2 numai daca distanta pana la eticheta destinatie nu depaseste 127 octeti
  execute a jump to the address determined by a2 only if the distance to the destination label is no more than 127 bytes

The correct answer is: semnala eroare de sintaxa
issue a syntax error

memory: FD FF

AX = 054A

add [x], 2
error: operation size not specified

mov ax, -1
mov bh, 1
idiv bh
Rezultatul este:
The result is:

Select one:

- a. Execution error ✖
- b. ah=00h; al=FFh
- c. Assembly error
- d. ax=0000h
- e. ah=00h; al=1999h

The correct answer is: ah=00h; al=FFh

AX = FF FF
BH = 01
idiv BH
-1 : 1 = -1 ⇒
AH = 00
AL = FF

Care este efectul executiei instructiunii "mov [a], -1" in conditiile definitiilor:
Which is the effect of the execution of instruction "mov [a],-1" for the following data definitions:
segment data
a resw 1
b db 3Ch, 4Dh

Select one:

- a. a=3Cffh
- b. eroare de sintaxa
  syntax error
- c. a=00ffh
- d. a=0ffffh ✖
- e. a=0ff3Ch

operation size not specified

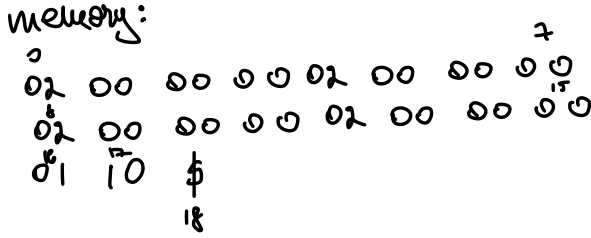
The correct answer is: eroare de sintaxa
syntax error

Ce valoare are contorul de locatii ($) la sfarsitul urmatoarelor declaratii de date (considerand ca offset-ul de inceput al segmentului de date este 0):
What will be the value of the location counter ($) at the end of the following data declaration (assuming that the starting offset of the segment is 0):
segment data ..
a times 4 dd 2
l equ 10
b dw a+1

Select one:

- a. 18 ✔
- b. 20
- c. 6
- d. 10

memory:



The correct answer is: 18

Care din instructiunile de mai jos sunt corecte sintactic?
Which of the following instructions are syntactically correct?
i) or word [a],2 ✓
ii) or [a],[a] → both operands from memory
iii) or 2,[a] → type?
iv) or 2,2 → both const + type?
v) or byte [a],2 | 5 ✓

0010 |
0101
0111 → 7

Select one:

- a. i), v) ✓
- b. Toate instructiunile specificate sunt corecte
  All mentioned instructions are correct
- c. Nici una din instructiunile specificate nu este corecta
  None of the mentioned instructions is correct
- d. iv)
- e. ii) ✖

The correct answer is: i), v)

Secventa de instructiuni:
mov ah, -128
mov bh, 80h
add ah,bh
seteaza flag-urile astfel:
The instructions sequence
mov ah, -128
mov bh, 80h
add ah,bh
sets the flag values in the following way:

Select one:

- a. SF=0 CF=1 OF=0 ZF=0
- b. SF=0 CF=1 OF=1 ZF=0
- c. SF=1 CF=0 OF=1 ZF=0
- d. SF=1 CF=1 OF=0 ZF=1 ✖
- e. SF=0 CF=1 OF=1 ZF=1 ✓

The correct answer is: SF=0 CF=1 OF=1 ZF=1

AH = 80h
BH = 80h
add AH, BH

CF = 1
OF = 1
ZF = 1
SF = 0

$$\begin{array}{r} 80 + \\ 80h \\ \hline 100h \end{array}$$

Dandu-se urmatorul segment de date:
Given the data segment below:
a times 2 dd 1234h
b db 16h
sa se precizeze ce valoare are octetul de la offset 6 (considerand ca offset-ul de inceput al segmentului este 0)
what value holds the octet at offset 6 (assuming that the starting offset of the segment is 0)

Select one:

- a. 00h ✓
- b. 34h
- c. 14h
- d. 12h ✖
- e. 16h

The correct answer is: 00h

memory:

0	1	2	3	4	5
34	12	00	00	34	12

6		
00	00	16

⇒ 00h

mov al, -2
mov bl, -128
imul bl
Rezultatul este:
The result is:

Select one:

- a. ax=FF00h ✖
- b. Assembly error
- c. ax=FFFFh
- d. ax=100b
- e. Execution error
- f. ax=100h ✓

The correct answer is: ax=100h

AL = FEh
BL = 80h
imul bl ⇒ $-2 * (-128) =$
$= 256 = 100h$

Care dintre urmatoarele seturi de instructiuni au toate acelasi efect asupra operandului destinatie ?
Which of the following set of instructions have all the same effect on the destination operand?

Select one:

- a. xor AX,1 ; not AX
- b. xor BX,BX; sub BX,BX; rcr BX,16;
- c. xor BX,BX; and BX,0; ror BX,16; sbb BX,BX
- d. nici unul din seturile enumerate
  none of the enumerated sets
- e. xor DX,DX; and AX,0; shl BX,16 ; sub BX,BX ✔

xor BX, BX BX=0
sub BX, BX BX=0
rcr BX, 16 BX=BX
and BX, 0 BX=0

xor DX, DX DX=0
and AX, 0 AX=0
shl BX, 16 BX=0
sub BX, BX BX=0

The correct answer is: xor DX,DX; and AX,0; shl BX,16 ; sub BX,BX

Valoarea contorului de locatii:
The value of the location counter:

Select one:

- a. este o adresa relativa la toate segmentele active, mai putin cel de stiva, acolo valoarea curenta fiind determinata de catre registrul ESP ✖
  is an address relative to all active segments, except the stack segment, where the current value is established by the value of the ESP register
- b. este un numar intreg desemnand lungimea curenta a segmentului activ
  is an integer number representing the current length of the active segment
- c. este un numar intreg relativ la segmentul de date activ si este folosit pentru calculul dimensiunii vectorilor definiti in program
  is an integer number relative to the active data segment and it is used for computing the size of the arrays defined in the program
- d. nici una dintre variantele specificate
  none of the specified answers
- e. este un numar intreg desemnand pozitia curenta in cadrul segmentului de cod activ
  is an integer number representing the current position in the active code segment

the location counter keeps track of the next available memory location that can be assigned within the current segment

The correct answer is: nici una dintre variantele specificate
none of the specified answers

Dandu-se urmatorul segment de date:
Given the data segment below:
a db 1, 2, 3, 10, 20, 30
sa se precizeze ce valoare are cuvantul de la offset 2
(considerand ca offset-ul de inceput al segmentului este 0)
what value holds the word at offset 2 (assuming that the starting offset of the segment is 0)

Select one:

- a. 3
- b. 103
- c. 2563 ✓
- d. 2010h
- e. 23
- f. A3h ✖

memory:
01 02 03 0A 14 1E
0 1 2

0A03h = $10 \cdot 16^2 + 3$
= 2560 + 3
= 2563

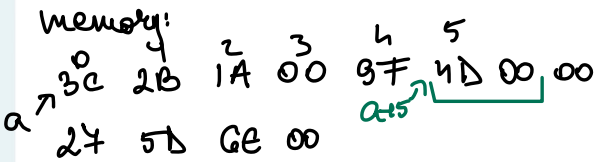
The correct answer is: 2563

Se da urmatorul segment de date:
The following data segment is given:
a dd 1a2b3ch, 4d9fh, 6e5d27h
Ce valoare va contine registrul BH in urma instructiunii:
What will be the value of BH after the execution of the instruction:
mov bx, [a+5]

Select one:

- a. 6eh
- b. eroare de sintaxa / syntax error
- c. d9h
- d. 0 ✓
- e. 9fh
- f. 4dh ✗

The correct answer is: 0



BX = 004Dh

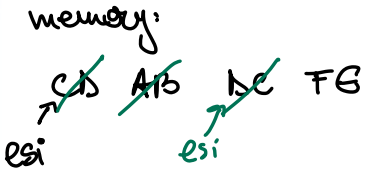
BH = 00

a dw 0ABCDh, 0FEDCh
---
mov esi, a
lodsw
lodsb
AL=?

Select one:

- a. 0ABh
- b. 0FEh
- c. 0CDh
- d. 0DCh ✓

The correct answer is: 0DCh



AL = DC

Care este efectul urmatoarei secvente de instructiuni?
Which is the effect of the following instructions sequence?
mov eax, -3 & -4; xor al,al; cbw; cwd;

Select one:

- a. eax=0ffffffffh
- b. eax=0fffffff9h
- c. eax=0ffff0000h ✓
- d. eax=00000000h
- e. eax=00000001h

The correct answer is: eax=0ffff0000h

-3 & -4

FF FF FF FD &
FF FF FF C
---
same C

1101 &
1100
1100

AL = 0

EAX : FF FF FF FC
FF FF FF 00
FF FF 00 00

DX : 0000

EAX : FF FF 00 00

Care este efectul executiei secventei:
What is the execution effect of the following sequence:
mov al, -1
neg al

Select one:

- a. CF=1; ZF=1;
- b. CF=1; SF=0; ✓
- c. CF=0; SF=1; ✗
- d. CF=0; ZF=1;

The correct answer is: CF=1; SF=0;

AL = FF ⇒ AL = 255

neg AL ⇒ AL = 01

neg computes the 2's comp.

⇒ SF = 0
ZF = 0
CF = 0 borrow from sub

Instructiunea cmp 0,0 este
The instruction cmp 0,0 is

Select one:

- a. eroare logica ✗
  logical error
- b. eroare de sintaxa ✓
  syntax error
- c. corecta
  correct

* invalid combination of operands

⇒ syntax error

The correct answer is: eroare de sintaxa
syntax error

Codul sursa de mai jos calculeaza corect expresia 600 / (-2) = -300, rest 0
Does the next code compute correctly the expression 600 / (-2) = -300, remainder 0
:

```
mov ax, 600
cwde
mov cx, -2
idiv cx
```

Select one:

- a. Nu ✓
  No
- b. Da
  Yes

ax = 600 = 258h
eax = 600 = 00 00 02 58h
cx = FEh
idiv cx ⇒ DX:AX / cx

⇒ incorect

The correct answer is: Nu
No

Formula de calcul a offset-ului unui operand se utilizeaza in:
The offset specification formula is used in\

Select one:

- a. adresarea indexata / indexed addressing
- b. adresarea directa si indirecta / direct and indirect addressing ✓
- c. adresarea directa / direct addressing
- d. adresarea indirecta / indirect addressing
- e. adresarea bazata / based addressing
- f. adresarea bazata si indexata / based and indexed addressing

The correct answer is: adresarea directa si indirecta / direct and indirect addressing

```
mov al, -2
mov bl, -128
mul al
```

Rezultatul este:
The result is:

Select one:

- a. ax=100b ✗
- b. ax=FFFFh
- c. Execution error
- d. ax=FC04h ✓
- e. ax=100h
- f. Assembly error

AL = -2 = FE
in unsigned int: 254
mul al = 254 · 254
= 64516

64516 | 16
1 | 4032 | 16
4 | 0 | 252 | 16
240 | 15 | 16
12 | 15 | 0
1
15

FC04

The correct answer is: ax=FC04h

Se da urmatorul segment de date:
The following data segment is given:
segment data
a db 20, 30
b resw 2
c dd 3
len equ ($-a)*8
Constanta "len" reprezinta numarul total de ..
The "len" constant represents the total number of ..

Select one:

- a. dublucuvinte din segmentul de date
  doublewords in the data segment
- b. octeti din segmentul de date
  bytes in the data segment
- c. cuvinte din segmentul de date
  words in the data segment
- d. biti din segmentul de date ✔
  bits in the data segment
- e. semiocteti din segmental de date
  nibbles in the data segment

The correct answer is: biti din segmentul de date
bits in the data segment

memory

1h 1E | | | | 03 00
00 00 $

len = ($-a)*8
= 10*8 = 80
biti in segmentul de date

Care este efectul executiei secventei:
What is the execution effect of the following sequence:
mov dh,62h
mov ch,200
sub dh,ch

Select one:

- a. CF=1; OF=1; ✓
- b. CF=0; OF=1; ✖
- c. CF=0; OF=0;
- d. CF=1; OF=0;

The correct answer is: CF=1; OF=1;

DH = 62h = 0110 0010
CH = 200 = C8h = 1100 1000

sub DH, CH

62 -
C8

9A h

CF = 1
0 - 1 = 1 => OF = 1

Care din urmatoarele instructiuni vor avea acelasi efect asupra registrului AL?
Which instructions will have the same effect on AL register?
i) mov al, 250 >>4; ii) mov al,0FFFFh>>4;
iii) mov al,0EFFFh>>12;
iv) mov al,-1>>4;
v) mov al,-1>>12;

Select one:

- a. i), iv)
- b. Fiecare din cele 5 secvente de instructiuni va avea un efect diferit asupra lui AL
  Each of the 5 sequences of instructions will have a different effect on AL
- c. i), ii), iv)
- d. ii), iii)
- e. Toate cele 5 secvente de instructiuni au acelasi efect
  All 5 sequences of instructions have the same effect
- f. ii), iv), v)

The correct answer is: ii), iv), v)

i) 250 >> 4 <=> FAh >> 4 <=>
1111 1010 >> 4 <=> 0000 1111 <=> 0Fh

ii) FFFF >> 4 <=> 0FFF

iii) EFFF >> 12 <=> 000E

iv) -1 >> 4 <=> FFFF
v) -1 >> 12 <=> FFFF
} signed!

Care este efectul executiei secventei:
What is the execution effect of the following sequence:
mov ax,400h
mov bl,0feh
idiv bl

Select one:
- a. CF=1; OF=1;
- b. No overflow
- c. Divide overflow ✔
- d. CF=0; OF=1;
- e. CF=1; OF=0;

AX = 0400h = 1024
BL = FEh = -2
idiv BL
AL ← 512 ∉ [-128, 127]
⇒ devide overflow

The correct answer is: Divide overflow

Ce valoare are contorul de locatii ($) la sfarsitul urmatoarelor declaratii de date (considerand ca
offset-ul de inceput al segmentului este 0):
Which is the value of the location counter ($) at the end of the following data declarations (assuming that the starting offset of the segment is 0):
segment data
a db '123','4','56'
b resd 1
c dw 4-1, 3

Select one:
- a. 14h
- b. 1010b
- c. 9h
- d. 0eh ✓
- e. 0bh ✖

memory:

0	1	2	3	4	5	6
31	32	33	34	35	36	␣
7	8	9	10	11	12	13
␣	␣	␣	03	00	03	00

$

14d = 0eh

The correct answer is: 0eh

Codul sursa de mai jos calculeaza corect expresia 65535/7
Does the next code compute correctly the expression 65535/7
mov ax, 65535
cwd
mov bx, 7
div bx

Select one:
- a. Da
  Yes
- b. Nu ✔
  No

AX = FF FF
DX:AX = FF FF FF FF
BX = 07
div BX DX:AX / BX
result doesn't fit on a word

The correct answer is: Nu
No

mov ax, -1
mov bh, 1
div bh
Rezultatul este:
The result is:

Select one:
- a. Execution error ✔
- b. ah=00h; al=FFh
- c. ax=0000h
- d. Assembly error
- e. ah=00h; al=1999h

AX = FF FF
BH = 01
div BH
result doesn't fit on a byte

Prin executia carei instructiuni/secvente de instructiuni EAX ramane nemodificat?
By running which instruction/sequence of instructions EAX remains the same?

Select one:

- a. prin ambele
  by both of them
- b. or EAX,EAX ✖
- c. mov cl,11h; rcr ax,cl
- d. prin niciuna
  by none of them

or EAX, EAX

CL = 17

rcr ax, cl

17 because we need to add the last value stored into CF back in AX

The correct answer is: prin ambele
by both of them

Registrii de adresa sunt:
The address registers are:

Select one:

- a. registrii generali, registri de segment şi EIP
  the general registers, the segment registers and EIP
- b. cei 4 registri de segment desemnand adresele de inceput ale segmentelor active din program si EIP ✖
  the 4 segment registers representing the starting addresses of the program's active segments and EIP
- c. toti registrii care participa la calculul de adresa efectuat de catre componenta ADR din BIU
  all the registers involved in the address computation performed by the ADR component from BIU
- d. registrii de baza si de index
  base and index registers
- e. toti registrii procesorului, deoarece toti sunt implicati in calculul de adresa
  all the processor's registers because all of them are involved in the address computation
- f. cei 6 registri de segment desemnand adresele de inceput ale segmentelor active din program si EIP
  the 6 segment registers representing the starting addresses of the program's active segments and EIP
- g. CS si EIP
  CS and EIP
- h. registrii de segment, EBP, ESP si EIP
  the segment registers, EBP, ESP and EIP
- i. CS, DS, SS, ES, EBX, ESI, EDI, ESP, EBP, EIP
- j. registrii de segment si EIP
  the segment registers and EIP

The correct answer is: registrii de segment si EIP
the segment registers and EIP

Se da urmatorul segment de date:
The following data segment is given:
a dd 0a344dh, 30h, 11223344h, 46ab89ch
Sa se precizeze ce valoare are dublucuvantul de la offset 9 (considerand ca offset-ul de inceput al segmentului este 0)
Which is the value of the doubleword from offset 9 (assuming that the starting offset of the segment is 0).

Select one:

- a. 46ab89ch
- b. 112246abh
- c. 46443322h
- d. 11223344h
- e. 9c112233h ✔

memory:

0	1	2	3	4	5	6
4D	34	0A	00	30	00	00
7	8	9				
00	44	33	22	11	9C	B8

6A 0h

⇒ 9C 11 22 33 h

The correct answer is: 9c112233h

Liniile urmatoare de cod calculeaza corect expresia a-b? :
Does the following code compute the expression a-b correctly? :

```
a dd 10000h
b dw 2
....
mov ax, [a]
mov dx, [a+2]
sub ax,[b]
sbb dx,0
```

Select one:

- a. da eroare la link-editare
  there is a linking error
- b. da eroare de executie
  there is an execution error
- c. nu da eroare, dar nu calculeaza corect valoarea expresiei ✖
  there is no error, but the expression value is not computed correctly
- d. da eroare la asamblare
  there is an assembly error
- e. da, calculeaza corect expresia
  yes, the expression is computed correctly ✓

The correct answer is: da, calculeaza corect expresia
yes, the expression is computed correctly

AX = 0000h

DX = 0001h

sub AX, [b]

↳ AX = FFFFEh

CF = 1

sbb DX, 0

↳ DX = 0000h

⇒ correct computation