1. The value of 00100111b is

27h

63h

26h

17h

1. The value of 0110\_0011b is

0D4

D4h

104d

063h

1. Which is the effect of the execution of instruction "mov eax, [b-1]" and „mov eax, [b+1]” if we have defined:

a dw 1A2Bh

b db 3Bh, 4Bh

c dd 1A2B3C4Dh;

2b 1a 3b 4b 4d 3c 2b 1a

Eax: 4d 4b 3b 1b

Eax: 1b 3c 4d 4b

1. a dw 1234h, 0aaaah

b db 2, 3, 0, 1

c dw 7, 8; 00 07 00 08

mov ax, [a+5]

mov bx, [a+3]

mov cx, [a+7]

mov dx, [b-2]

34 12 aa aa 02 03 00 01 07 00 08 00

1. What is the effct of; „b equ 123456h” in memory? How many bytes are allocated?
2. What does the following instr? mov al, (3&4)^(13^(\~13); !, << ,>>

& =AND

^ = XOR

~ = Complement to one

| = OR

! = NOT

% = reminder unsigned

%% = reminder signed

<< = shift

>> = shift

/ = unsigned div

// = signed div

1. What is the value of the $ after the following defs:.

0 a dd 0ab123456dfh

b times 3 dw 1ah

$

c times 6 db 'd'

$

4+ 6 + 6

1. What is the value of the $ after the following defs:

a dw 0ab1210ab10h

b times 2 dw 1ah, 0hDc; 0hAC = 0ACh= 0ACH = 0ACx

c equ 6

d times 6 db 'b'

10 ab 1a 00 dc 00 1a 00 dc 00 62 62 62 62 62 62

1. mov al,16

mov bl,10

mul bl

ax=160

ax=160h

ax=00a0h

ax=FFF0h

Assembly error

Execution error

1. mov al,3

mov bl,-3

imul bl

ax=90

ax=0009h

ax=FF09h

ax=FFF7h

Assembly error

Execution error

1) -9 => 0000\_1001 => c to 1 => 1111\_0110 => cto 2 (+1)

2) 16-9 => 7 ultima cifra in hexa FF F7

-3 => 16-3 = 13 = Dh => FF FD

-19 => 16 + 16-19 = 13 = Dh => FF ED

-35 => 2\*16 +16-35 = 13 =Dh => FF DD

1. mov al,3

mov bl, -3

mul al

ax?

1. mov al,3

mov bl,-3

mul bl

ax=FD \*3

ax=0009h

ax=FF09h

ax=02F7h

Assembly error

Execution error

1. What will I have in ch?

mov ch, -1 rol 6;

-1 = FF = 1111\_1111 =>

Sintax err

1. Which instructions are correct:

not byte [a],3

not byte [a], byte [a]

not 3, byte [a]

not byte [a],3 & 6

not 3,3

1. Will the jump be executed?

a db 'abcd'

b db 'abcd'

cld

mov esi, a

mov edi, b

cmpsb

cmpsb

JB eticheta

1. Mov ah,0

Mov ax, 256; 256d = 100h; ax: 0100

Mov bl,1;; bl =1

Div bl; ax: bl =ah-rest; al=cat

0100: 1= 100h -> al = XX!!! Overflow

1. Mov ah,0

Mov al, 256; 256d=100h 2 al=00h

; Mov al, 256h = al=56h

Mov bl,1;

Div bl

1. Mov ah,0

Mov ax, 256;

Div byte 1 -> err sintax, nu se pot inmultiri/impartiri cu constanta!

1. Mov ah,0; ah=0

Mov ax, 22h; ax = 00 22;

Mov dx,0; dx = 00 00;

Mov cx, 3 ; cx= 00 03

Div cx ; (dx:ax) 22 / 3 => ax: 7 DX=1

Mov cx, 2; cx =00 02

Div cx ; DX:ax = 00 01 00 07: 2 => 10007:2

Ax =?

20. Care sunt valorile resultate?

a db 1,2,3,4,5; - 5

b db 1,2,3,4,5; - 5

c dw 1,2,3,4,5 ; - 10

$

lenA1 equ $-a; 20

lenA2 equ $-b; 15

lenA3 equ $-c; 10

lenB2 equ $-c/2; 100400 - 100400/2 = 50xxx!!!

lenB2 equ ($-c)/2; = 5

d dd 1,2,3,4,5 ;

lenB4 equ ($-d); = 4\* 5elem

lenB4a equ ($-d)/4

lenB5 equ $-b – c; = err

21. what will be the result of the following instr:

mov al, 128 ; 128d = 80 = 1000\_0000

cbw => ax: 1111\_1111\_1000\_0000 = FF 80: 88 40

mov bl,2; bl =2

div bl;

- al = 40

- al = 40d

- al = 40,5

- execution error - T (overflow)

- sintax error, cbw should not be used there

- al = 40h

- al = 0100\_0000

22.

Which one of the follwing address(es) is/are valid?

ss, ds, cs, es: baza + index \* scale + numar; scale: 0, 1, ,2, 4,8

a) mov eax, [eax\*9 + 12] ;=> eax + 8\*eax + 12

b) mov eax, [ss:ebx + eax + 3] =>

c) mov eax, [esi + 2\*esp + 1] =care e registrul care nu poate fi folosit ca index?

d) mov eax, [esp + 2\*esi + 1]

e) mov eax, [ebx + 3 \* eax + 1]

f) mov eax, [edx + 4 \* eax + 2]

g) mov eax, [edx + 9 \* eax]

h) mov eax, [edx + eax \* 9]

23. Ce valoare va genera urm cod?

|  |  |
| --- | --- |
| mov ax,61440  mov bl,5  div bl | mov ax,61440  mov bl,5  idiv bl |

* Primul caz – check for overflow
* Al doilea caz pt ca e imul, considera ca in ax e numar negativ –

**Facute anterior:**

1. Which of the following variable definitions are correct?

(H or X for hexadecimal, D or T for decimal, Q or O for base 8 and B or Y for binary)

* a0 db 10b;
* a1 dw 0h12;
* a2 db 123456h;
* a3 db 1111\_0011\_1111\_0000b;
* \_a4 dw 124h;
* a5# db 33h;
* ?a6 dw 233h;
* a7~ db 1233h;

1. Are the following definitions equivalent?

a)

* + b0 db 256;
  + b1 dd 100h;

1. * b2 db 1203h;
   * b3 db 3;
2. Do the instructions generate the same result?

v db 4;

x db 5;

* + mov ax, 4;
  + mov ax, v;

d)

v db 4;

x db 5;

* + mov ax, 4;
  + mov eax, [v];

e)

v db 4;

add ax, 5;

* + add v, 5;
  + add [v], 5;