

Fie codul de mai jos. Care sunt valorile lui **eax** si **edx** cand executia ajunge la label? *

Un punct

```
.data
    x: .long 17
    y: .long 6
.test
.global main
main:
    mov $1, %edx
    mov x, %eax
    jmp et
    mov $0, %edx
et:
    divl y
label:
    mov $1, %eax
    mov $0, %ebx
    int $0x80
```

- ☒ **eax = 0x2aaaaaad, edx = 0x3**
- ☐ **eax = 0x2, edx = 0x5**
- ☐ **eax = 0x5, edx = 0x2**
- ☐ **eax = 0x3, edx = 0x2aaaaaad**

Se stocheaza in EAX valoarea 0x80000000, in EBX 0x8, in ECX 0x1 si in EDX 0x4. Ce valori vor avea registrii EAX, respectiv EDX dupa executarea instructiunii **mul %ebx**? *

Un punct

- ☒ **eax = 0, edx = 4**
- ☐ **eax = 32, edx = 0**
- ☐ **eax = 32, edx = 4**
- ☐ **eax = 4, edx = 0**

Se stocheaza in %edx valoarea 0, in eax 37 si in ebx 15. Ce valori vor avea registrii eax * 2 puncte si edx dupa executarea instructiunii **div %ebx** ?

- ☐ eax = 7, edx = 2
- ☐ eax = 7, edx = 0
- ☐ eax = 2, edx = 0
- ☒ eax = 2, edx = 7

Fie codul de mai jos. Care este valoarea depozitata in **z** cand ajungem la eticheta **final**? * 2 puncte

```
.data
x: .long 17
y: .long 6
x1: .long 5
y1: .long 9
z: .space 4
.text
.global main
main:
    mov x, %eax
    mov y, %ebx

    cmp %eax, %ebx
    jge et1

    mov x1, %eax
    cmp %eax, %ebx
    jle et

    mov x, %ebx
    sub %eax, %ebx
    jmp final

et:
    add %eax, %ebx
    jmp final

et1:
    mov x1, %eax
    mov y1, %ebx
    cmp %eax, %ebx
    jge et2
    add %eax, %ebx
    jmp final

et2:
    sub %eax, %ebx

final:
    mov %ebx, z

    mov $1, %eax
    mov $0, %ebx
    int $0x80
```

- ☐ 4
- ☐ 1
- ☒ 12
- ☐ 23

Care este ordinea de trecere prin etichete? *

2 puncte

```
.data
.text
.global main
main:
    jmp etb
eto:
    jmp etd
eth:
    jmp eto
etb:
    jmp eth
etd:
    mov $1, %eax
    mov $0, %ebx
    int $0x80
```

- ☒ etb, eth, eto, etd
- ☐ eth, etb, etd, eto
- ☐ eto, eth, etb, etd
- ☐ etb, eto, eth, etd

De cate ori se va executa instructiunea **loop**? *

Un punct

```
.data
x: .long 5
y: .long 5
s: .long 0
.text
.global main
main:
    mov x, %ecx
    sub y, %ecx

et:
    add %ecx, s
    loop et
exit:
    mov $1, %eax
    mov $0, %ebx
    int $0x80
```

- ☐ 0x1
- ☐ 0x0
- ☐ 0x5
- ☒ 0xffffffff
- ☐ este un ciclu infinit

Fie urmatorul program. Ce valoare vom obtine daca vom rula cu debuggerul urmatoarele comenzi?

* Un punct

b et_exit

run

i r edx

```
.data
.text
.global main
main:
    mov $2, %eax
    mov $3, %ebx
    add %eax, %ebx
    mul %ebx
    mov $0, %edx
    divl $3
    add %eax, %edx
et_exit:
    mov $1, %eax
    mov $0, %ebx
    int $0x80
```

☐ 0x1

☒ 0x4

☐ 0x0

☐ 0x3

Acest conținut nu este nici creat, nici aprobat de Google.

Formulare Google