Ce valori vor fi depozitate in v cand executia va ajunge in dreptul etichetei **et_exit** ? * Un punct

```
.data
  v: .space 20
  n: .long 5
.text
.global main
main:
  lea v, %edi
  mov $11, %edx
  mov $0, %ecx
et_loop:
  cmp n, %ecx
  jg et_exit
  mov %edx, (%edi, %ecx, 4)
  inc %ecx
  inc %edx
  jmp et_loop
et_exit:
  mov $1, %eax
  xor %ebx, %ebx
  int $0x80
```

- ○ 11, 12, 13, 14, 15, 16
- ○ 11, 12, 13, 14, 15
- ◉ 11, 12, ... , 27
- ○ 0, 1, 2, 3, 4, 5
- ○ Executia nu ajunge la et_exit, loop-ul este infinit.

Ce se va afisa pe ecran? *

2 puncte

```
.data
  x:  .long 1, 3, 6, 7, 9
  n1: .long 5
  n2: .long 10
  c:  .long 0x64636261
  s:  .space 11
.text
.global main
main:
  mov $s, %edi
  mov $x, %esi
  movb c, %al
  mov $0, %ecx

et_loop1:
  cmp n2, %ecx
  je et_exit_loop1
  mov %al, (%edi, %ecx, 1)
  inc %ecx
  jmp et_loop1

et_exit_loop1:
  mov $c, %eax
  movb 1(%eax), %al
  mov $0, %ecx
et_loop2:
  cmp n1, %ecx
  je et_exit
  mov (%esi, %ecx, 4), %ebx
  mov %al, (%edi, %ebx, 1)
  inc %ecx
  jmp et_loop2

et_exit:
  mov $10, %ecx
  movb $0, (%edi, %ecx,1)
  mov $4, %eax
  mov $1, %ebx
  mov $s, %ecx
  mov $11, %edx
  int $0x80

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

- (•) ababaabbab
- ( ) 61, 61, 61, 61, 61, 61, 61, 61, 61, 61
- ( ) aaaaaaaaaa
- ( ) bbbbbaaaaa

Ce se va afisa pe ecran? *

2 puncte

```
.data
  n: .long 3
  s: .asciz "abc"
.text
.global main
main:
  mov $s, %edi
  mov $0, %ecx
et_loop:
  cmp n, %ecx
  je et_exit
  mov (%edi, %ecx, 1), %al
  sub $'a', %al
  add $'A', %al
  mov %al, (%edi, %ecx, 1)
  inc %ecx
  jmp et_loop

et_exit:
  mov $4, %eax
  mov $1, %ebx
  mov $s, %ecx
  mov $4, %edx
  int $0x80

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

- ( ) abc
- ( ) Abc
- (•) ABC
- ( ) Abc + o valoarea reziduala

Ce se va afisa pe ecran? *

Un punct

```
.data
  n: .long 3
  s: .byte 'a', 'b', 'c'
  t: .byte 'd', 'e', 'f'
  u: .space 4
.text
.global main
main:
  mov $0, %ecx
et_loop:
  cmp n, %ecx
  je et_exit
  mov $0, %edx
  sub %ecx, %edx
  mov t(, %edx, 1), %al
  mov %al, u(, %ecx, 1)
  inc %ecx
  jmp et_loop

et_exit:
  movb $0, u(, %ecx, 1)

  mov $4, %eax
  mov $1, %ebx
  mov $u, %ecx
  mov $4, %edx
  int $0x80

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

- ○ def
- ○ abc
- ○ cba
- ◉ dcb

Fie urmatorul program. Ce valoare vom obtine daca vom rula cu debuggerul urmatoarele comenzi? * 2 puncte

b et_exit
run
i r eax

```
.data
  n: .long 4
  v: .long 0x01020304, 0x05060708, 0x090a0b0c, 0x0d0e0f10
.text
.global main
main:
  mov $v, %esi
  mov $1, %ecx
  mov (%esi, %ecx, 4), %eax
  add $4, %esi
  movb (%esi, %ecx, 4), %al
et_exit:
  mov $1, %eax
  xor %ebx, %ebx
  int $0x80
```

- (●) 0x0506070c
- ( ) 0x090a0b0c
- ( ) 0x05060704
- ( ) 0x090a0b08

Fie urmatorul program. Ce valoare vom obtine daca vom rula cu debuggerul urmatoarele comenzi? * 2 puncte

b et_exit
run
i r eax

```
.data
  n: .long 4
  v: .long 0x01020304, 0x05060708, 0x090a0b0c, 0x0d0e0f10
.text
.global main
main:
  mov $v, %esi
  mov $2, %ecx
  mov -8(%esi, %ecx, 4), %eax
et_exit:
  mov $1, %eax
  xor %ebx, %ebx
  int $0x80
```

- (●) 0x01020304
- ( ) 0x090a0b0c
- ( ) 0x05060708
- ( ) Executia nu va ajunge la et_exit, o sa apara o eroare de accesare