

Topics covered in [Lecture 1-Lecture 9], Lecture 12, Lecture 13

## Written Exam Structure

**Office:** 1 point, working time: 100 mins

### 1. Memory representation and register values – 3 points

#### 1a. Memory representation of a data segment (1.5 points):

- Each variable (those defined with DB, DW, DD, DQ, or RESB, RESW, RESQ) is represented in memory depending on its type.
- The base used for representation is base 16 (hexadecimal).
- Variables are represented at the byte level.
- For compound data types (words, doublewords, quadwords), bytes are represented from right to left (in reverse order of their writing/appearance).
- Constants do not appear in memory (constants are those defined with EQU).
- Characters occupy one byte and are represented by their ASCII code.

#### 1b. Memory representation of a code segment and register values (1.5 points):

2. Problem with expression (signed/unsigned) – 2.5 points with comments

3. Problem with strings (all types of strings) – 2.5 points, with comments

4. Exercise with instruction comparison or filling blanks – 1 point, with code examples

Test Example

1a. Represent the next Data segment in Memory (hex dump)

a db 12

b dw -12

c dd 210

d dq 12ABh

e equ 10

f db 'x,y'

g equ \$-f

G1 db \$-f ; g1 = 3 (byte)

1b. Show the Memory representation or register/variables values (line by line) for the next code segment

mov ax, 1234h

mov bx, 12345h

mov esi, a

mov edi, d

cld

lodsw

mov cx, ax

mov cl, -1

mov ah, cl

Stosw

Mov dl, g

Mov dl, g1 ; eroare dl byte, g1 adresa (adresele sunt pe dd)

2. Write a code sequence that computes, in signed interpretation:

$$\text{Exp} = a / (-f) + b * c - d + e$$

Where:

- a is a byte
- b is a word
- c is a byte
- d is a doubleword
- e is a quadword
- f is a byte

3. A string of doublewords is given. Identify and save in another string all the bytes divisible by 7 from the initial string (o secv repetitia si o secv de test cmp)

4.

Exemplu a. Compare by identifying similarities and differences between the MOVSW instruction and the MOV instruction.

Mov – transfer instr

Movsw instr speciala de lucru cu siruri

A db 3

Mov al, [a]

Exemplu 1

S db 1,2,3, 4

D db 10, 11, 12, 13

Mov esi, S

Mov edi, D

Cld ; sau std

Movsw ; mov word[d+edi], word[s+esi], add esi,2, add edi, 2 (cld) sau sub esi, 2, sub edi, 2 (std)

S in mem: 01 02 03 04

0 1 2 3

Esi

01 02

D in mem: 0a 0b 0c 0d

0 1 2 3

Esi

Exemplu 2:

S db 1,2,3, 4

D dw 10, 11, 12, 13

Mov esi, S

Mov edi, D

Cld ; sau std

Movsw ; mov word[d+edi], word[s+esi], add esi,2, add edi, 2 (cld) sau sub esi, 2, sub edi, 2 (std)

S in mem: 01 02 03 04

0 1 2 3

Esi

01 02

D in mem: 0a 0b 0c 0d 00

0 1 2 3

Esi

Exemplu b. Compare the AND instruction and the TEST instruction.

And modifica op destinatie

Test face and dar nu modifica nici un operand

Mov al, 00001111b

Mov bl, 11110000b

And al, bl ; al = 0      test al, bl ; rez = 0 dar al si bl raman neschimbati