

Saptamana 4

1. Raspundeti la urmatoarele intrebari:

A. Explicati functionarea urmatoarelor instructiuni de deplasare si rotire:

- a. SHL, SAL, SHR, SAR
- b. ROR, ROL, RCR, RCL

B. Explicati functionarea urmatoarelor instructiuni de salt:

- a. CALL, RET
- b. JMP
- c. LOOP
- d. JC, JE, JZ, JS
- e. JA, JB
- f. JG, JL

C. Explicati functionarea instructiunilor de I/E: IN, OUT

D. Explicati functionarea urmatoarelor instructiuni pe siruri:

- a. MOVSB, MOVSW
- b. LODSB, LODSW, STOSB si STOSW
- c. REP

Instructiuni de deplasare

2. Problema rezolvata in s4ex1.asm

Sa se implementeze in limbaj de asamblare expresia de mai jos folosind instructiuni de deplasare:

$$AX = 7*AX + 2*BX + BX/8$$

- a. Se va compila s4ex1.asm
- b. Se va executa in OllyDbg
- c. Se vor urmari schimbarile care au loc la nivelul registrilor si a flagurilor

3. Sa se implementeze in limbaj de asamblare expresia de la punctul 2 folosind instructiuni aritmetice (MUL si DIV).

Adresarea memoriei

4. Problema rezolvata in s4ex2.asm

Sa se scrie un program care copiaza un sir de valori din locatii consecutive de memorie in alta locatie, in ordine inversa, prin intermediul stivei.

- a. Se va compila fisierul sursa
- b. Se va executa in OllyDbg
- c. Se vor urmari schimbarile care au loc la nivelul registrilor, memoriei si a flagurilor

5. Sa se scrie un program care calculeaza media unui sir de numere intregi din memorie.

Numerele sunt de tip octet. Media va fi memorata ca valoare intreaga intr-o variabila de tip octet.

6. Problema rezolvata in s4ex3.asm

Sa se scrie un program pentru adunarea a doua matrice bidimensionale.

- a. Se va compila fisierul sursa
- b. Se va executa in OllyDbg
- c. Se vor urmari schimbarile care au loc la nivelul registrilor, memoriei si a flagurilor

Salturi

7. Implementarea unei structuri similare cu IF-THEN-ELSE in limbaj de asamblare:

Pseudocod:

```
if (AX > BX) then AX = AX+1
                else BX = BX+1
```

Limbaj de asamblare (varianta pentru numere cu semn):

```
CMP AX, BX      ;se compara cei doi registri
JG et_then
INC BX          ;daca BX >=AX
JMP et_iesire   ;daca s-a executat ramura òelseò, nu se va mai executa ramura òthenò
                ;se sare peste

et_then:
INC AX          ;daca AX>BX
et_iesire:
```

8. Implementarea unei structuri de tip FOR in limbaj de asamblare:

Pseudocod:

```
for (i = 0 ; i < n ; i ++)  
{  
    AX = AX+1  
}
```

Limbaj de asamblare:

Varianta 1

```
MOV DI, n  
MOV SI, 0    ; corespondentul lui i  
et_for:  
INC AX  
INC SI      ;incrementam SI  
CMP SI, DI  
JBE et_for
```

Varianta 2

```
MOV CX, n    ;CX contorizeaza bucla realizata de LOOP  
et_loop:  
INC AX  
LOOP et_loop ;CX = CX-1, (CX == 0)?
```

9. Problema rezolvata in s4ex4.asm

Sa se determine minimul si maximul dintr-un sir de numere fara semn reprezentate pe octet si sa se scrie valorile gasite in memorie.

- a. Se va compila fisierul sursa
- b. Se va executa in OllyDbg
- c. Se vor urmari schimbarile care au loc la nivelul registrilor, memoriei si a flagurilor

10. Sa se determine minimul si maximul dintr-un sir de numere cu semn reprezentate pe cuvint si sa se scrie valorile gasite in memorie.