

**Principi programiranja.
Konsultacije za završni ispit**

Polazne veličine su $a = 1$, $b = 2$, $c = 3$, biće
 $y_1 = \text{FF80}$, $y_2 = 0006$, $y_3 = 01\text{BA}$, hex.

**Zadaci (1–12).
Programi na jeziku asemblera za Intel
8086/Pentium**

1. Izračunati $y_1 = 2ab + 1$, $y_2 = y_1 - 2$, $y_3 = y_2 - 2$, $y_4 = y_3 - 16$, po dva bajta, debug.exe

Rješenje:

ulazni podaci: $a = (10)_{16}$, $b = (20)_{16}$, rezultati:
 $y_1 = 401$, $y_2 = 3\text{FF}$, $y_3 = 3\text{FD}$, $y_4 = 3\text{ED}$, hex.

Držimo a na 200 (a na 200 i 201), b na 202,
 y_1 na 204, y_2 na 206, y_3 na 208, y_4 na 20A.

```
A 100 (direktiva assemble, unosimo naredbe)
→ 100 MOV AX, [200] (AX ← c(200))
    MOV CX, [202] (CX ← c(202))
    MUL CX (množenje AX ← AX · CX)
    SHL AX, 1 (shift left AX)
    INC AX (AX ← AX + 1)
    MOV [204], AX (c(204) ← AX)
    SUB AX, 2 (AX ← AX - 2)
    MOV [206], AX (c(206) ← AX)
    SUB AX, 2 (AX ← AX - 2)
    MOV [208], AX (c(208) ← AX)
    SUB AX, 10 (AX ← AX - 10 hex)
    MOV [20A], AX (c(20A) ← AX)
    INT 20 (stop)
A 200 (assemble, unosimo brojeve)
200 DW 10 (a = 10 hex)
202 DW 20 (b = 20 hex)
G =100 (direktiva go, da počne izvršavanje)
{Program terminated normally}
D 200 20B (direktiva dump, šta piše u memoriji)
10 00 20 00 01 04 FF 03-FD 03 ED 03
Q (direktiva quit, izlazimo iz programa)
```

(Iz A assemble izlazi se sa "enter". Pomoću
A 100 loadovali smo naredbe, a pomoću A 200
podatke)

Vidimo da je dump saopštio 0010 0020 0401
03FF 03FD 03ED

2. Izračunati $y_1 = -16a - 20b - 24c$, $y_2 = a^2 + ab + ac$, $y_3 = \frac{y_1^2+1}{y_2^2+1}$

Rješenje: plan memorije:

200	202	204	206	208	20A
a	b	c	y_1	y_2	y_3

```
→ 100 MOV AX, [200] (AX ← c(200))
    MOV CX, 10 (CX ← 10 hex)
    MUL CX (AX ← AX · CX)
    MOV BX, AX (BX ← AX)
    MOV AX, [202] (AX ← c(202))
    MOV CX, 14 (CX ← 14 hex)
    MUL CX (AX ← AX · CX)
    ADD BX, AX (BX ← BX + AX)
    MOV AX, [204] (AX ← c(204))
    MOV CX, 18 (CX ← 18 hex)
    MUL CX (AX ← AX · CX)
    ADD BX, AX (BX ← BX + AX)
    NEG BX (BX ← -BX, sada je BX = y1)
    MOV [206], BX (ostavljam y1 na 206)
    MOV AX, [200] (AX ← c(200))
    ADD AX, [202] (AX ← AX + c(202))
    ADD AX, [204] (AX ← AX + c(204))
    MOV CX, [200] (CX ← c(200))
    MUL CX (AX ← AX · CX, sada je AX = y2)
    MOV [208], AX (c(208) ← AX)
    MOV AX, [206] (AX ← c(206))
    MOV CX, [206] (CX ← c(206))
    MUL CX (AX ← AX · CX)
    INC AX (AX ← AX + 1)
    MOV [20A], AX (c(20A) ← AX)
    MOV AX, [208] (AX ← c(208))
    MOV CX, [208] (CX ← c(208))
    MUL CX (AX ← AX · CX)
    INC AX (AX ← AX + 1)
    MOV BX, AX (BX ← AX)
    MOV AX, [20A] (AX ← c(20A))
    XOR DX, DX (DX ← 0)
    DIV BX (AX ← AX/BX, sada je AX = y3)
    MOV [20A], AX (ostavljam y3 na 20A)
    INT 20 (stop)
200 DW 1 (a = 1)
202 DW 2 (b = 2)
204 DW 3 (c = 3)
G =100 (go, da počne izvršavanje)
D 200 20B (dump, šta piše u memoriji)
01 00 02 00 03 00 80 FF-06 00 BA 01
```

Dump saopštava da je (nakon svega) a , b , c ,
 y_1 , y_2 , $y_3 = 0001 0002 0003 \text{FF80 } 0006 01\text{BA}$.

MUL reg znači $\text{AX} \leftarrow \text{AX} \cdot \text{reg}$ ili tačnije (DX
 $\text{AX}) \leftarrow \text{AX} \cdot \text{reg}$. DIV reg znači $\text{AX} \leftarrow \text{AX} / \text{reg}$
ili tačnije $\text{AX} \leftarrow (\text{DX } \text{AX}) / \text{reg}$.

3. Stek, upisati $x_1 = 11$, $x_2 = 12$, $x_3 = 13$, $x_4 = 14$, izračunati $y = x_1 + 2x_2 + 3x_3 + 4x_4$ čitanjem sa steka.

Rješenje:

```

→ 100 MOV SP,2F8    (stack pointer ← 2F8)
      MOV AX,B      (AX ← B)
      PUSH AX       (pošalji AX na stek)
      MOV AX,C      (AX ← C)
      PUSH AX       (pošalji AX na stek)
      MOV AX,D      (AX ← D)
      PUSH AX       (pošalji AX na stek)
      MOV AX,E      (AX ← E)
      PUSH AX       (pošalji AX na stek)
      XOR BX,BX     (BX ← 0)
      POP AX        (skini sa steka, stavi u AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      POP AX        (skini sa steka, stavi u AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      POP AX        (skini sa steka, stavi u AX)
      ADD BX,AX     (BX ← BX + AX)
      ADD BX,AX     (BX ← BX + AX)
      POP AX        (skini sa steka, stavi u AX)
      ADD BX,AX     (BX ← BX + AX)
      MOV [200],BX  (c(200) ← BX)
      INT 20        (stop)
G =100    (go, da počne izvršavanje)
D 2F0 2F7 (dump, šta piše u memoriji)
OE 00 0D 00 0C 00 0B 00
D 200 201 (dump, šta piše u memoriji)
82 00     (znači da je c(200) = 82, c(201) = 0)

```

Nakon svih naredbi "push", članovi steka zauzimaju pozicije 2F6, 2F4, 2F2 i 2F0 redom.

Veličinu y držimo na poziciji 200 u memoriji. Vidimo da je $y = 82$ hex ($y = 130$ dekadno).

4. Stek, treba upisati $x_1 = 11$, $x_2 = 12$, $x_3 = 13$, $x_4 = 14$ i zatim izračunati $y = x_1^2 + x_2^2$ čitanjem sa steka.

Rješenje:

```

→ 100 MOV SP,2F8    (stack pointer ← 2F8)
      MOV AX,B      (AX ← B)
      PUSH AX       (pošalji AX na stek)

```

```

      MOV AX,C      (AX ← C)
      PUSH AX       (pošalji AX na stek)
      MOV AX,D      (AX ← D)
      PUSH AX       (pošalji AX na stek)
      MOV AX,E      (AX ← E)
      PUSH AX       (pošalji AX na stek)
      POP AX        (skini sa steka, stavi u AX)
      POP AX        (skini sa steka, stavi u AX)
      POP AX        (skini sa steka, stavi u AX)
      MOV CX,AX     (CX ← AX)
      MUL CX        (AX ← AX · CX)
      MOV BX,AX     (BX ← AX)
      POP AX        (skini sa steka, stavi u AX)
      MOV CX,AX     (CX ← AX)
      MUL CX        (AX ← AX · CX)
      ADD BX,AX     (BX ← BX + AX)
      MOV [200],BX  (c(200) ← BX)
      INT 20        (stop)
G =100    (go, da počne izvršavanje)
D 200 201 (dump, šta piše u memoriji)
09 01     (vidimo da je c(200) = 9, c(201) = 1)
D 2F0 2F7 (čemu su jednaki c(2F0), ..., c(2F7))
OE 00 0D 00 0C 00 0B 00

```

U programu debug.exe ima više tzv. assembler-skih direktiva. Jedna od njih je D, od riječi dump. Njena sintaksa je D $a_1 a_2$ a značenje je: prikazati na ekranu sadržaj dijela memorije od adrese a_1 do adrese a_2 , sve jedan po jedan bajt, znači adrese $a_1, a_1 + 1, \dots, a_2$.

Promjenljivu y držimo na adresi 200. Vidimo da je na kraju $y = 109$ hex ($y = 265$).

5. Učitati preko tastature četiri slova (upisati ih u memoriju), prekid INT 21, funkcija AH = 1 (ovo je u heksadekadnom). Zatim štampati u prvom redu učitana slova, a u drugom redu ta ista slova, prekid INT 21, funkcija AH = 2, CR + LF = prelazak na početak novog reda.

Rješenje:

Ako se ekran nalazi u tekstualnom modu i ako mu se uputi znak Carriage Return (ascii D) onda će se pozicija vratiti na početak (istog) reda. Ako se uputi Line Feed (ascii A) onda se pozicija pomjeri za jedan red naniže, bez vraćanja ulijevo. Slična su dešavanja sa matričnim štampačem kada mu se pošalju dva karaktera D i A: tekuća pozicija za štampanje postaviće se na početak idućeg reda.

```

→ 100 MOV AH,1      (AH ← 1)
      INT 21        (poziva se rutina)

```

```

MOV [200],AL (c(200) ← AL)
INT 21 (poziva se rutina)
MOV [201],AL (c(201) ← AL)
INT 21 (poziva se rutina)
MOV [202],AL (c(202) ← AL)
INT 21 (poziva se rutina)
MOV [203],AL (c(203) ← AL)
MOV AH,2 (AH ← 2)
MOV DL,[200] (DL ← c(200))
INT 21 (poziva se rutina)
MOV DL,[201] (DL ← c(201))
INT 21 (poziva se rutina)
MOV DL,[202] (DL ← c(202))
INT 21 (poziva se rutina)
MOV DL,[203] (DL ← c(203))
INT 21 (poziva se rutina)
MOV DL,D (DL ← D)
INT 21 (poziva se rutina)
MOV DL,A (DL ← A)
INT 21 (poziva se rutina)
MOV DL,[200] (DL ← c(200))
INT 21 (poziva se rutina)
MOV DL,[201] (DL ← c(201))
INT 21 (poziva se rutina)
MOV DL,[202] (DL ← c(202))
INT 21 (poziva se rutina)
MOV DL,[203] (DL ← c(203))
INT 21 (poziva se rutina)
INT 20 (stop)
G =100 (go, da počne izvršavanje)

```

otkucati ceti (bez enter)

na ekranu izlazi:

```

ceti
ceti

```

Ako → 100 MOV AH,8 onda nema odjeka prilikom kucanja.

AX	AH	AL
----	----	----

6. Učitati preko tastature četiri slova, a zatim dvaput štampati učitane četvorku slova pomoću prekida INT 21, funkcija AH = 9 čiji je smisao: štampanje niza slova (prikazivanje niza slova na ekran).

Rješenje:

Slova se učitavaju pomoću INT 21, funkcija AH = 1. Što se tiče funkcije AH = 9: DX – gdje počinje niz slova, \$ = ascii (24)₁₆ za kraj niza slova.

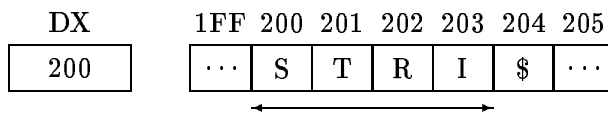
```

→ 100 MOV AH,1 (AH ← 1, funkcija)
INT 21 (poziva se rutina)
MOV [200],AL (c(200) ← AL)
INT 21 (poziva se rutina)
MOV [201],AL (c(201) ← AL)
INT 21 (poziva se rutina)
MOV [202],AL (c(202) ← AL)
INT 21 (poziva se rutina)
MOV [203],AL (c(203) ← AL)
MOV AL,24 (AL ← 24 hex)
MOV [204],AL (c(204) ← AL)
MOV AH,9 (AH ← 9, funkcija)
MOV DX,200 (DX ← 200)
INT 21 (poziva se rutina)
INT 21 (poziva se rutina)
INT 20 (stop)
G =100 (go, da počne izvršavanje)

```

otkucati stri

na ekranu će se dopisati stristri



$$7. y = \begin{cases} 2a + 2b, & \text{ako je } c < 0 \\ 8a + 8b, & \text{ako je } c \geq 0 \end{cases}$$

Rješenje:

200	202	204	206
a	b	c	y

```

→ 100 MOV AX,[200] (AX ← c(200))
ADD AX,[202] (AX ← AX + c(202))
SHL AX,1 (shift left AX tj. AX ← 2 · AX)
MOV [206],AX (c(206) ← AX)
MOV BX,[204] (BX ← c(204))
CMP BX,0 (compare, uporedi BX i 0)
JL 11C (ako je BX < 0 onda skoči na 11C)
SHL AX,1 (shift left AX tj. AX ← 2 · AX)
SHL AX,1 (shift left AX tj. AX ← 2 · AX)
MOV [206],AX (c(206) ← AX)
11C INT 20 (stop)
200 DW 7 (a = 7)
202 DW 8 (b = 8)
204 DW 9 (c = 9)
G =100 (go, da počne izvršavanje)
D 200 207 (dump, šta piše u memoriji)
07 00 08 00 09 00 78 00 (y = 78 hex)

```

Malo promijenimo:

204 DW FEA9 (c negativan broj)
 G =100 (go, da počne izvršavanje)
 D 200 207 (dump, šta piše u memoriji)
 07 00 08 00 A9 FE 1E 00 (sada y = 1E hex)

D 200 205 (dump, šta piše u memoriji)
 05 00 0A 00 33 00 (vidimo da je y = 33 hex)

U zadacima je dozvoljeno da se u sličnim situacijama umjesto 11C, ima na dva mjesta u programu (kao JMP 11C i kao 11C INT 20), napiše label ili nešto slično.

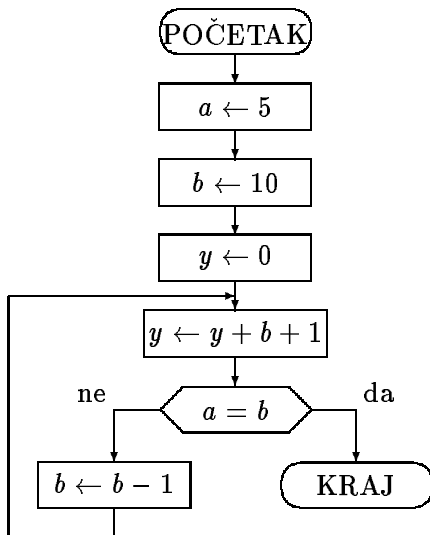
$$9. y = \sum_{k=a}^b (k + 1)$$

Rješenje:

200	202	204	AX	BX	CX
a	b	y	a	b	y

CMP d, s znači if d = s then goto lab
 JE lab

Mogućnosti su JE d = s, JNE d ≠ s, JG d > s,
 JGE d ≥ s, JL d < s i JLE d ≤ s



→ 100 MOV AX, [200] (AX ← c(200))
 MOV BX, [202] (BX ← c(202))
 MOV CX, 0 (CX ← 0)
 10A ADD CX, BX (CX ← CX + BX)
 INC CX (CX ← CX + 1)
 CMP AX, BX (uporedi AX i BX)
 JE 114 (ako AX = BX onda 114)
 DEC BX (BX ← BX - 1)
 JMP 10A (skok na 10A)
 114 MOV [204], CX (c(204) ← CX)
 INT 20 (stop)
 200 DW 5 (a = 5 hex)
 202 DW A (b = A hex)
 G =100 (go, da počne izvršavanje)

11. Rad sa nizom (indirektno adresiranje):
 treba izračunati zbir 16 brojeva (po dva bajta),
 nalaze se od 1000 do 101F, $y = \sum_{i=1}^{16} x_i$.

Nije dozvoljeno da se izvrši 16 naredbi
 ADD kao rješenje! Treba kao niz.

Rješenje:

Registar AX služi za sumu. Registar BX služi
 kao indeks-registar. Rezultat y ostavljamo na 200
 (dva bajta).

→ 100 XOR AX, AX AX ← 0
 102 MOV BX, 1000 BX ← 1000
 105 ADD AX, [BX] AX ← AX + c(BX)
 107 CMP BX, 101E uporedi AX i 101E
 10B JE 111 skoči ako su jednaki
 10D INC BX BX ← BX + 1
 10E INC BX BX ← BX + 1
 10F JMP 105 skoči na 105
 111 MOV [200], AX c(200) ← AX
 114 INT 20 stop
 1000 DW 0 $x_1 = c(1000) = 0$
 1002 DW 2 $x_2 = c(1002) = 2$
 1004 DW 4 $x_3 = c(1004) = 4$
 1006 DW 6 $x_4 = c(1006) = 6$
 1008 DW 8 $x_5 = c(1008) = 8$
 100A DW A $x_6 = c(100A) = A$
 100C DW C x_7
 100E DW E x_8
 1010 DW 10 x_9
 1012 DW 12 x_{10}
 1014 DW 14 x_{11}
 1016 DW 16 x_{12}
 1018 DW 18 x_{13}
 101A DW 1A x_{14}
 101C DW 1C $x_{15} = c(101C) = 1C$
 101E DW 1E $x_{16} = c(101E) = 1E$
 G =100 go, da počne izvršavanje
 D 200 201 dump, šta piše u memoriji
 F0 00 vidimo da je c(200) = F0, c(201) = 00

Izračunali smo da je $0 + 2 + 4 + \dots + 30 = 240$
 dekadno.

$$8. y_1 = a^2 + b^2 - 100, y_2 = \text{sgn } y_1$$

$$10. y = \sum_{k=a}^b (-1)^k k$$

12. Najveći među 16 brojeva, pomoću indirektnog adresiranja.