

Ministerul Educației, Culturii și Cercetării Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatica și Microelectronică

RAPORT

Lucrare de laborator Nr.3
la Arhitectura Calculatoarelor

Tema: Arhitectura calculatoarelor
Tema: „**Programe liniare**”

A efectuat:

Victoria Popescu

A verificat:

Colesnic V.

Chișinău 2019

Scopul lucrării:

Lucrarea prezintă instrucțiunile pentru transferuri de date, instrucțiuni în aritmetica binară și în aritmetica BCD.

Mersul lucrării:

Condiție: Conform variantei elaborați 2 variante de program :

- 1) cu introducerea datelor de la tastatură și afișarea rezultatelor pe ecran.
- 2) cu generarea datelor de intrare, utilizând procedurile Random32, RandomRange și afișarea rezultatelor pe ecran.

$$13 \quad \left| \quad z = \begin{cases} (Y - 2X)/5 + 150, & \text{dacă } X/2 > Y \\ 2X - 47 + Y & , \text{dacă } X/2 \leq Y \end{cases} \right.$$

1.1 Codul sursa varianta 13 introducerii de la tastatura

```
INCLUDE Irvine32.inc
.data
mes1 byte "Introduceti valoarea X:",0
mes2 byte "Introduceti valoarea Y:",0
mes3 byte "Rezutatul este:",0
vrx dword 0
vry dword 0
interm dw ?
rez dd 0

.code
main PROC
mov edx,OFFSET mes1
call WriteString ; afisarea mes1
call ReadDec ; introducerea de la tastatura
mov vrx,eax ; salvarea valorii in variabila vrx

mov edx,OFFSET mes2
call WriteString ; afisarea mes2
call ReadDec ; introducerea de la tastatura
mov vry,eax ; salvarea valorii in variabila vry

;controlam conditiile
xor eax,eax
mov edx,0

mov eax, vrx
mov bx, 2
div bx ; calcul X/2
cmp vry, eax ;compararea Y cu X/2
jb con1 ; salt la con1, daca Y < x/2

mov eax, vrx ;realizam expresia (y-2x)/5 + 150 daca nu se realizeaza jb la con1
mov bx,2
mul bx
mov ebx, vry
sub ebx, eax
mov eax, ebx
mov bx, 5
cwd
idiv bx
add eax, 150
mov rez, eax
jmp ex ; salt neconditionat la ex
```

```

con1: mov eax, vrx ; realizam expresia 2*X - 47 + Y , daca x<=y/2
mov bx, 2
mul bx
sub eax, 47
add eax, vry
mov rez,eax
ex: mov edx,OFFSET mes3
call WriteString ; afisarea mesajului mes3
call WriteInt ; afisarea rezultatului cu semn
call CrLf ; din rand nou

        exit
main ENDP
END main

```

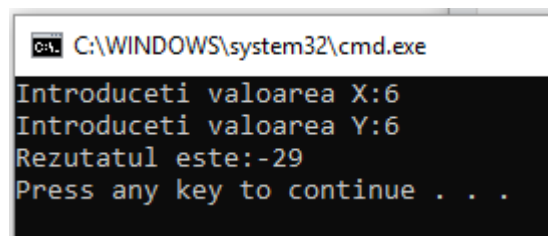


Fig.2 Execuția programului 4 varianta introducerii de la tastatură

1.2 Codul sursa varianta 13 metoda variabilelor x si y random

```

; 2*X - 47 + Y , daca Y < X/2
; (y-2x)/5 + 150 , daca Y >= X/2

INCLUDE Irvine32.inc

.data
mes1 byte "Introduceti valoarea X:",0
mes2 byte "Introduceti valoarea Y:",0
mes3 byte "Rezutatul este:",0
vrx  dword 0
vry  dword 0
interm dw ?
rez  dd 0

.code

main PROC
call Randomize
mov edx,OFFSET mes1
call WriteString ; afisarea mes1
call Random32 ; apelarea functiei random
call WriteInt ; afisarea lui X la ecran
call CrLf
mov vrx,eax ; salvarea valorii in variabila vrx

mov edx,OFFSET mes2
call WriteString ; afisarea mes2
call Random32 ; apelarea functiei random
call WriteInt ; afisarea lui Y la ecran
call CrLf

```

```

mov vry,eax ; salvarea valorii in variabila vry

;controlam conditiile
xor eax,eax
mov edx,0

mov eax, vrx
mov bx, 2
div bx ; calcul X/2
cmp vry, eax ;compararea Y cu X/2
jb con1 ; salt la con1, daca Y < x/2

mov eax, vrx ;realizam expresia (y-2x)/5 + 150 daca nu se realizeaza jb la con1
mov bx,2
mul bx
mov ebx, vry
sub ebx, eax
mov eax, ebx
mov bx, 5
cwd
idiv bx
add eax, 150
mov rez, eax
jmp ex ; salt neconditionat la ex

con1: mov eax, vrx ; realizam expresia 2*X - 47 + Y , daca Y < X/2
mov bx, 2
mul bx
sub eax, 47
add eax, vry
mov rez,eax
ex: mov edx,OFFSET mes3
call WriteString ; afisarea mesajului mes3
call WriteInt ; afisarea rezultatului cu semn
call Crlf ; din rand nou

    exit
main ENDP
END main

```

```

C:\WINDOWS\system32\cmd.exe
Introduceti valoarea X:-1442681048
Introduceti valoarea Y:-973495220
Rezutatul este:+1878819437
Press any key to continue . . .

```

Fig.2 Execuția programului 13 metoda variabilelor random

Listing-ul programului varianta 13 metoda introducerii de la tastatura:

```

1.    INCLUDE Irvine32.inc
2.                                C ; Include file for Irvine32.lib                (Irvine32.inc)

```

```

3.          C
4.          C ;OPTION CASEMAP:NONE          ; optional: make identifiers case-sensitive
5.          C
6.          C INCLUDE SmallWin.inc          ; MS-Windows prototypes, structures, and
constants
7.          C .NOLIST
8.          C .LIST
9.          C
10.         C INCLUDE VirtualKeys.inc
11.         C ; VirtualKeys.inc
12.         C .NOLIST
13.         C .LIST
14.         C
15.         C
16.         C .NOLIST
17.         C .LIST
18.         C
19.         00000000          .data
20.         00000000 49 6E 74 72 6F      mes1 byte "Introduceti valoarea X:",0
21.             64 75 63 65 74
22.             69 20 76 61 6C
23.             6F 61 72 65 61
24.             20 58 3A 00
25.         00000018 49 6E 74 72 6F      mes2 byte "Introduceti valoarea Y:",0
26.             64 75 63 65 74
27.             69 20 76 61 6C
28.             6F 61 72 65 61
29.             20 59 3A 00
30.         00000030 52 65 7A 75 74      mes3 byte "Rezutatul este:",0
31.             61 74 75 6C 20
32.             65 73 74 65 3A
33.             00
34.         00000040 00000000          vrx  dword 0
35.         00000044 00000000          vry  dword 0
36.         00000048 00000000          rez  dd 0
37.
38.         00000000          .code
39.         00000000          main PROC
40.         00000000 BA 00000000 R      mov edx,OFFSET mes1
41.         00000005 E8 00000000 E      call WriteString ; afisarea mes1
42.         0000000A E8 00000000 E      call ReadDec ; introducerea de la tastatura
43.         0000000F A3 00000040 R      mov vrx,eax ; salvarea valorii in variabila vrx
44.
45.         00000014 BA 00000018 R      mov edx,OFFSET mes2
46.         00000019 E8 00000000 E      call WriteString ; afisarea mes2
47.         0000001E E8 00000000 E      call ReadDec ; introducerea de la tastatura
48.         00000023 A3 00000044 R      mov vry,eax ; salvarea valorii in variabila vry
49.
50.             ;controlam conditiile
51.         00000028 33 C0          xor eax,eax
52.         0000002A BA 00000000      mov edx,0
53.
54.         0000002F A1 00000044 R      mov eax, vry
55.         00000034 66 BB 0002      mov bx, 2
56.         00000038 66 F7 F3      div bx ; calcul Y/2
57.         0000003B 39 05 00000040 R cmp vrx, eax ;compararea X cu y/2
58.         00000041 72 2B          jb con1 ; salt la con1, daca X < y/2
59.
60.         00000043 A1 00000040 R      mov eax, vrx ;realizam expresia (y-2x)/5 + 150 daca nu se
realizeaza jb la con1
61.         00000048 66 BB 0002      mov bx,2
62.         0000004C 66 F7 E3      mul bx
63.         0000004F 8B 1D 00000044 R mov ebx, vry
64.         00000055 2B D8          sub ebx, eax
65.         00000057 8B C3          mov eax, ebx

```

```

66.      00000059 66| BB 0005      mov bx, 5
67.      0000005D 66| 99          cwd
68.      0000005F 66| F7 FB          idiv bx
69.      00000062 05 00000096      add eax, 150
70.      00000067 A3 00000048 R      mov rez, eax
71.      0000006C EB 1A          jmp ex ; salt neconditionat la ex
72.
73.      0000006E A1 00000040 R      con1: mov eax, vrx ; realizam expresia 2*X - 64 + Y , daca
x<=y/2
74.      00000073 66| BB 0002      mov bx, 2
75.      00000077 66| F7 E3          mul bx
76.      0000007A 83 E8 40      sub eax, 64
77.      0000007D 03 05 00000044 R      add eax, vry
78.      00000083 A3 00000048 R      mov rez,eax
79.
80.      00000088 BA 00000030 R      ex: mov edx,OFFSET mes3
81.      0000008D E8 00000000 E      call WriteString ; afisarea mesajului mes3
82.      00000092 E8 00000000 E      call WriteInt ; afisarea rezultatului cu semn
83.      00000097 E8 00000000 E      call Crlf ; din rand nou
84.
85.                      exit
86.      0000009C 6A 00          *      push  +000000000h
87.      0000009E E8 00000000 E      *      call  ExitProcess
88.      000000A3                      main ENDP
89.                      END main

```

Listing-ul programului varianta 13 metoda random x si y:

```

1.      INCLUDE Irvine32.inc
2.                      C ; Include file for Irvine32.lib              (Irvine32.inc)
3.                      C
4.                      C ;OPTION CASEMAP:NONE                ; optional: make identifiers case-sensitive
5.                      C
6.                      C INCLUDE SmallWin.inc                ; MS-Windows prototypes, structures, and
constants
7.                      C .NOLIST
8.                      C .LIST
9.                      C
10.                     C INCLUDE VirtualKeys.inc
11.                     C ; VirtualKeys.inc
12.                     C .NOLIST
13.                     C .LIST
14.                     C
15.                     C
16.                     C .NOLIST
17.                     C .LIST
18.                     C
19.      00000000          .data
20.      00000000 58 3D 00      mes1 byte "X=",0
21.      00000003 59 3D 00      mes2 byte "Y=",0
22.      00000006 52 65 7A 75 74      mes3 byte "Rezutatul este:",0
23.      61 74 75 6C 20
24.      65 73 74 65 3A
25.      00
26.      00000016 00000000      vrx  dword 0
27.      0000001A 00000000      vry  dword 0
28.      0000001E 00000000      rez  dd 0
29.
30.      00000000          .code

```

```

31.      00000000          main PROC
32.      00000000 E8 00000000 E      call Randomize
33.      00000005 BA 00000000 R      mov edx,OFFSET mes1
34.      0000000A E8 00000000 E      call WriteString ; afisarea mes1
35.      0000000F E8 00000000 E      call Random32 ; apelarea functiei random
36.      00000014 E8 00000000 E      call WriteInt ; afisarea lui X la ecran
37.      00000019 E8 00000000 E      call Crlf
38.      0000001E A3 00000016 R      mov vrx,eax ; salvarea valorii in variabila vrx
39.
40.      00000023 BA 00000003 R      mov edx,OFFSET mes2
41.      00000028 E8 00000000 E      call WriteString ; afisarea mes2
42.      0000002D E8 00000000 E      call Random32 ; apelarea functiei random
43.      00000032 E8 00000000 E      call WriteInt ; afisarea lui Y la ecran
44.      00000037 E8 00000000 E      call Crlf
45.      0000003C A3 0000001A R      mov vry,eax ; salvarea valorii in variabila vry
46.
47.          ;controlam conditiile
48.      00000041 33 C0          xor eax,eax
49.      00000043 BA 00000000      mov edx,0
50.
51.      00000048 A1 0000001A R      mov eax, vry
52.      0000004D 66| BB 0002      mov bx, 2
53.      00000051 66| F7 F3          div bx ; calcul Y/2
54.      00000054 39 05 00000016 R cmp vrx, eax ;compararea X cu y/2
55.      0000005A 72 2B          jb con1 ; salt la con1, daca X < y/2
56.
57.      0000005C A1 00000016 R      mov eax, vrx ;realizam expresia (y-2x)/5 + 150 daca nu se
realizeaza jb la con1
58.      00000061 66| BB 0002      mov bx,2
59.      00000065 66| F7 E3          mul bx
60.      00000068 8B 1D 0000001A R mov ebx, vry
61.      0000006E 2B D8          sub ebx, eax
62.      00000070 8B C3          mov eax, ebx
63.      00000072 66| BB 0005      mov bx, 5
64.      00000076 66| 99          cwd
65.      00000078 66| F7 FB          idiv bx
66.      0000007B 05 00000096      add eax, 150
67.      00000080 A3 0000001E R      mov rez, eax
68.      00000085 EB 1A          jmp ex ; salt neconditionat la ex
69.
70.      00000087 A1 00000016 R      con1: mov eax, vrx ; realizam expresia 2*X - 64 + Y , daca
x<=y/2
71.      0000008C 66| BB 0002      mov bx, 2
72.      00000090 66| F7 E3          mul bx
73.      00000093 83 E8 40      sub eax, 64
74.      00000096 03 05 0000001A R add eax, vry
75.      0000009C A3 0000001E R      mov rez,eax
76.
77.      000000A1 BA 00000006 R      ex: mov edx,OFFSET mes3
78.      000000A6 E8 00000000 E      call WriteString ; afisarea mesajului mes3
79.      000000AB E8 00000000 E      call WriteInt ; afisarea rezultatului cu semn
80.      000000B0 E8 00000000 E      call Crlf ; din rand nou
81.
82.          exit
83.      000000B5 6A 00          *      push +00000000h
84.      000000B7 E8 00000000 E      *      call ExitProcess
85.      000000BC          main ENDP
86.          END main

```

Concluzie:

În urma efectuării lucrării de laborator nr. 3 s-a continuat familiarizarea cu operațiile aritmetice punându-le în practica rezolvării a unei probleme liniare care realizează verificare, după care în funcție de condiția care o satisface, execută una din ecuații cu două necunoscute, din sistem. S-au utilizat mai multe tipuri de date – db, dd, dw, etc. Totodată s-au utilizat și instrucțiuni precum add, mov, imul, div, etc. și s-au pus în practică cunoștințele referitor la modurile de adresare a operanzilor. În concluzie s-a familiarizat cu limbajul Assembler.