**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 3

la cursul ***„Arhitectura calculatoarelor”***

***Varianta 4***

**A efectuat:**  **St. gr. CR-221FR Serba Cristina**

**A verificat: lect. univ. Postovan Dumitru**

**Chișinău 2024**

# Scopul lucrării:

Lucrarea prezinta instructiunile pentru transferuri de date, instructiuni in aritmetica binara si in aritmetica BCD.

# Codurile sursă:

1. cu introducerea datelor de la tastatură şi afişarea rezultatelor pe ecran.

|  |
| --- |
| INCLUDE Irvine32.inc  .data  mes1 byte "Introduceti valoarea X:",0  mes2 byte "Introduceti valoarea Y:",0  mes3 byte "Rezultatul este:",0  vrx dword 0  vry dword 0  rez dd 0  .code  main PROC  ; Afisare mesaj pentru X  mov edx, OFFSET mes1  call WriteString  call ReadDec  mov vrx, eax ; Salvare valoare X  ; Afisare mesaj pentru Y  mov edx, OFFSET mes2  call WriteString  call ReadDec  mov vry, eax ; Salvare valoare Y  ; Calculam X > Y / 2  mov eax, vrx  mov ebx, vry  shr ebx, 1 ; Y / 2  cmp eax, ebx  jg con1 ; Daca X > Y / 2, mergem la con1  ; Daca X <= Y / 2, calculam Z = 2X - 64 + Y  mov eax, vrx  shl eax, 1 ; 2X  add eax, vry  sub eax, 64  mov rez, eax  jmp ex ; Sarim la afisare rezultat  con1:  ; Daca X > Y / 2, calculam Z = (Y - 2X) / 5 + 150  mov eax, vrx  mov ebx, vry  shl ebx, 1 ; 2Y  sub eax, ebx  mov ebx, 5  xor edx, edx ;Resetam registrul edx pentru a nu avea overflow  div ebx ; (Y - 2X) / 5  add eax, 150  mov rez, eax  ex:  ; Afisare rezultat  mov edx, OFFSET mes3  call WriteString  call WriteInt ; Afisare rezultat  call Crlf ; Caractere de sfarsit de linie  exit  main ENDP  END main |

1. cu generarea datelor de intrare, utilizănd procedurile Random32, RandomRange:

|  |
| --- |
| INCLUDE Irvine32.inc  .data  mes1 byte "Valoarea generata pentru X este:",0  mes2 byte "Valoarea generata pentru Y este:",0  mes3 byte "Rezultatul este:",0  vrx dword 0  vry dword 0  rez dd 0  .code  main PROC  ; Generare valoare random pentru X  call RandomRange  mov vrx, eax  mov edx, OFFSET mes1  call WriteString  mov eax, vrx  call WriteInt  call Crlf  ; Generare valoare random pentru Y  call RandomRange  mov vry, eax  mov edx, OFFSET mes2  call WriteString  mov eax, vry  call WriteInt  call Crlf  ; Calculam X > Y / 2  mov eax, vrx  mov ebx, vry  shr ebx, 1 ; Y / 2  cmp eax, ebx  jg con1 ; Daca X > Y / 2, mergem la con1  ; Daca X <= Y / 2, calculam Z = 2X - 64 + Y  mov eax, vrx  shl eax, 1 ; 2X  add eax, vry  sub eax, 64  mov rez, eax  jmp ex ; Sarim la afisare rezultat  con1:  ; Daca X > Y / 2, calculam Z = (Y - 2X) / 5 + 150  mov eax, vrx  mov ebx, vry  shl ebx, 1 ; 2Y  sub eax, ebx  mov ebx, 5  xor edx, edx ; Resetam registrul edx pentru a nu avea overflow  div ebx ; (Y - 2X) / 5  add eax, 150  mov rez, eax  ex:  ; Afisare rezultat  mov edx, OFFSET mes3  call WriteString  mov eax, rez  call WriteInt ; Afisare rezultat  call Crlf ; Caractere de sfarsit de linie  exit  main ENDP  END main |

# Fișierele listing:

1. cu introducerea datelor de la tastatură şi afişarea rezultatelor pe ecran:

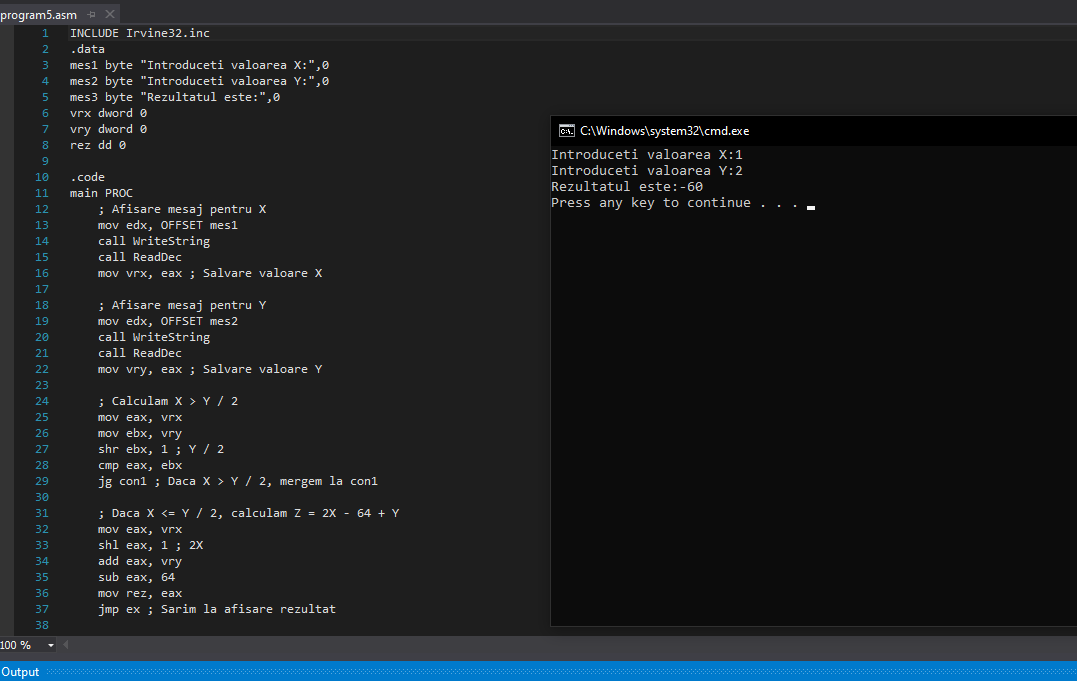
|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 19:57:04  program5.asm Page 1 - 1  INCLUDE Irvine32.inc  C ; Include file for Irvine32.lib (Irvine32.inc)  C  C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive  C  C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants  C .NOLIST  C .LIST  C  C INCLUDE VirtualKeys.inc  C ; VirtualKeys.inc  C .NOLIST  C .LIST  C  C  C .NOLIST  C .LIST  C  00000000 .data  00000000 49 6E 74 72 6F mes1 byte "Introduceti valoarea X:",0  64 75 63 65 74  69 20 76 61 6C  6F 61 72 65 61  20 58 3A 00  00000018 49 6E 74 72 6F mes2 byte "Introduceti valoarea Y:",0  64 75 63 65 74  69 20 76 61 6C  6F 61 72 65 61  20 59 3A 00  00000030 52 65 7A 75 6C mes3 byte "Rezultatul este:",0  74 61 74 75 6C  20 65 73 74 65  3A 00  00000041 00000000 vrx dword 0  00000045 00000000 vry dword 0  00000049 00000000 rez dd 0  00000000 .code  00000000 main PROC  ; Afisare mesaj pentru X  00000000 BA 00000000 R mov edx, OFFSET mes1  00000005 E8 00000000 E call WriteString  0000000A E8 00000000 E call ReadDec  0000000F A3 00000041 R mov vrx, eax ; Salvare valoare X  ; Afisare mesaj pentru Y  00000014 BA 00000018 R mov edx, OFFSET mes2  00000019 E8 00000000 E call WriteString  0000001E E8 00000000 E call ReadDec  00000023 A3 00000045 R mov vry, eax ; Salvare valoare Y  ; Calculam X > Y / 2  00000028 A1 00000041 R mov eax, vrx  0000002D 8B 1D 00000045 R mov ebx, vry  00000033 D1 EB shr ebx, 1 ; Y / 2  00000035 3B C3 cmp eax, ebx  00000037 7F 17 jg con1 ; Daca X > Y / 2, mergem la con1  ; Daca X <= Y / 2, calculam Z = 2X - 64 + Y  00000039 A1 00000041 R mov eax, vrx  0000003E D1 E0 shl eax, 1 ; 2X  00000040 03 05 00000045 R add eax, vry  00000046 83 E8 40 sub eax, 64  00000049 A3 00000049 R mov rez, eax  0000004E EB 22 jmp ex ; Sarim la afisare rezultat  00000050 con1:  ; Daca X > Y / 2, calculam Z = (Y - 2X) / 5 + 150  00000050 A1 00000041 R mov eax, vrx  00000055 8B 1D 00000045 R mov ebx, vry  0000005B D1 E3 shl ebx, 1 ; 2Y  0000005D 2B C3 sub eax, ebx  0000005F BB 00000005 mov ebx, 5  00000064 33 D2 xor edx, edx ;Resetam registrul edx pentru a nu avea overflow  00000066 F7 F3 div ebx ; (Y - 2X) / 5  00000068 05 00000096 add eax, 150  0000006D A3 00000049 R mov rez, eax  00000072 ex:  ; Afisare rezultat  00000072 BA 00000030 R mov edx, OFFSET mes3  00000077 E8 00000000 E call WriteString  0000007C E8 00000000 E call WriteInt ; Afisare rezultat  00000081 E8 00000000 E call Crlf ; Caractere de sfarsit de linie  exit  0000008D main ENDP  END main  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 19:57:04  program5.asm Symbols 2 - 1 |

1. cu generarea datelor de intrare, utilizănd procedurile Random32, RandomRange:

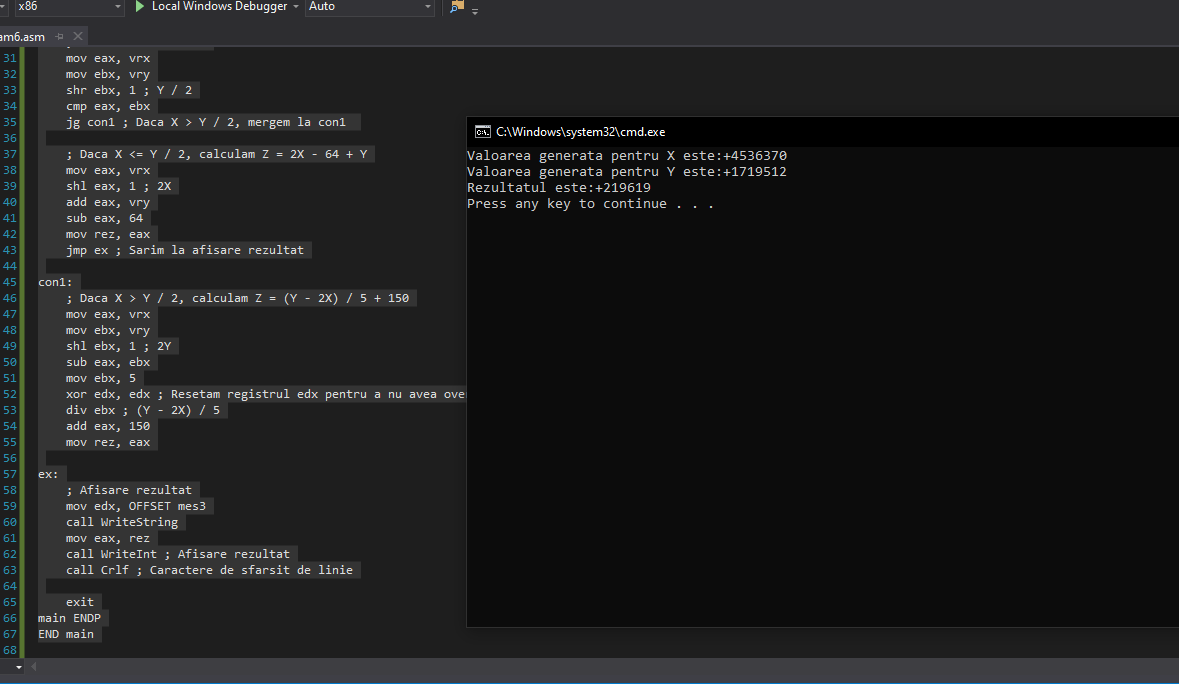
|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 19:55:20  program6.asm Page 1 - 1  INCLUDE Irvine32.inc  C ; Include file for Irvine32.lib (Irvine32.inc)  C  C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive  C  C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants  C .NOLIST  C .LIST  C  C INCLUDE VirtualKeys.inc  C ; VirtualKeys.inc  C .NOLIST  C .LIST  C  C  C .NOLIST  C .LIST  C  00000000 .data  00000000 56 61 6C 6F 61 mes1 byte "Valoarea generata pentru X este:",0  72 65 61 20 67  65 6E 65 72 61  74 61 20 70 65  6E 74 72 75 20  58 20 65 73 74  65 3A 00  00000021 56 61 6C 6F 61 mes2 byte "Valoarea generata pentru Y este:",0  72 65 61 20 67  65 6E 65 72 61  74 61 20 70 65  6E 74 72 75 20  59 20 65 73 74  65 3A 00  00000042 52 65 7A 75 6C mes3 byte "Rezultatul este:",0  74 61 74 75 6C  20 65 73 74 65  3A 00  00000053 00000000 vrx dword 0  00000057 00000000 vry dword 0  0000005B 00000000 rez dd 0  00000000 .code  00000000 main PROC  ; Generare valoare random pentru X  00000000 E8 00000000 E call RandomRange  00000005 A3 00000053 R mov vrx, eax  0000000A BA 00000000 R mov edx, OFFSET mes1  0000000F E8 00000000 E call WriteString  00000014 A1 00000053 R mov eax, vrx  00000019 E8 00000000 E call WriteInt  0000001E E8 00000000 E call Crlf  ; Generare valoare random pentru Y  00000023 E8 00000000 E call RandomRange  00000028 A3 00000057 R mov vry, eax  0000002D BA 00000021 R mov edx, OFFSET mes2  00000032 E8 00000000 E call WriteString  00000037 A1 00000057 R mov eax, vry  0000003C E8 00000000 E call WriteInt  00000041 E8 00000000 E call Crlf  ; Calculam X > Y / 2  00000046 A1 00000053 R mov eax, vrx  0000004B 8B 1D 00000057 R mov ebx, vry  00000051 D1 EB shr ebx, 1 ; Y / 2  00000053 3B C3 cmp eax, ebx  00000055 7F 17 jg con1 ; Daca X > Y / 2, mergem la con1  ; Daca X <= Y / 2, calculam Z = 2X - 64 + Y  00000057 A1 00000053 R mov eax, vrx  0000005C D1 E0 shl eax, 1 ; 2X  0000005E 03 05 00000057 R add eax, vry  00000064 83 E8 40 sub eax, 64  00000067 A3 0000005B R mov rez, eax  0000006C EB 22 jmp ex ; Sarim la afisare rezultat  0000006E con1:  ; Daca X > Y / 2, calculam Z = (Y - 2X) / 5 + 150  0000006E A1 00000053 R mov eax, vrx  00000073 8B 1D 00000057 R mov ebx, vry  00000079 D1 E3 shl ebx, 1 ; 2Y  0000007B 2B C3 sub eax, ebx  0000007D BB 00000005 mov ebx, 5  00000082 33 D2 xor edx, edx ; Resetam registrul edx pentru a nu avea overflow  00000084 F7 F3 div ebx ; (Y - 2X) / 5  00000086 05 00000096 add eax, 150  0000008B A3 0000005B R mov rez, eax  00000090 ex:  ; Afisare rezultat  00000090 BA 00000042 R mov edx, OFFSET mes3  00000095 E8 00000000 E call WriteString  0000009A A1 0000005B R mov eax, rez  0000009F E8 00000000 E call WriteInt ; Afisare rezultat  000000A4 E8 00000000 E call Crlf ; Caractere de sfarsit de linie  exit  000000B0 main ENDP  END main  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 19:55:20  program6.asm Symbols 2 - 1 |

Rezultatul executării programului:

1. cu introducerea datelor de la tastatură şi afişarea rezultatelor pe ecran:



1. cu generarea datelor de intrare, utilizănd procedurile Random32, RandomRange:



# Concluzii:

În urma efectuării lucrării de laborator, am implementat un program în limbajul assembly care calculează o expresie condițională utilizând proceduri din biblioteca Irvine. Programul demonstrează utilizarea eficientă a registrelor, operațiilor aritmetice și instrucțiunilor de control al fluxului (cmp, jg, jmp) pentru a gestiona logica de ramificare. Generarea automată a datelor de intrare cu ajutorul RandomRange a permis testarea flexibilă pe un set variat de valori, fără intervenția utilizatorului. De asemenea, am observant că există cazul în care poate apărea excepție de integer overflow, care a fost rezolvată prin resetarea registrului edx.