**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Tema: Programare Liniare**

Lucrare de laborator nr.3

la disciplina Arhitectura Calculatoarelor

**Student gr. TI-173: Heghea Nicolae**

**Conducător: Colesnic Victor**

# Scopul lucrării

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

# Conditia

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.

Varianta 9 :

# Realizarea Conditiei 1 “Citirea de la tastieră”

Codul programului.

|  |
| --- |
| INCLUDE Irvine32.inc  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  .data  mes1 db " Valoarea X : ",0  mes2 db " Valoarea Y : ",0  mes3 db " Rezutatul este : ",0  mes4 db " Conditia 1 : ",0  mes5 db " Conditia 2 : ",0  mes6 db " EDX = ",0  mes7 db " EAX = ",0  mes8 db " Rest = ",0  vrx dd 0  vry dd 0  vrz dd -15  rez1 dd 0  rez2 dd 0  rez3 dd 0  .code  main PROC  call Randomize  lea edx, mes1 ; incarcam adresa mes1 in edx  call WriteString ; afisarea mes1  call ReadInt  mov vrx, eax ; salvarea valorii in variabila vrx  lea edx, mes2  call WriteString ; afisarea mes2  call ReadInt  mov vry, eax ; salvarea valorii in variabila vry  mov eax, vrx ; incarcam x in eax  mov ebx, 2 ; incarcam 2 in ebx  cdq  idiv ebx ; impartim x la 2  cmp eax, vry ; compararea x cu eax  jge con2 ; daca >= cond 2    con1:  lea edx, mes4  call WriteString ; afisam mes4  call Crlf  mov edx, 0  mov eax, vry ; incarcam vry in eax  cdq ; extindem eax  sub eax, 56  jnc salt1  cdq ; extindem daca scaderea este cu imprumut  salt1:  mov ebx, vrz ; incarcam -15  idiv ebx ; impartim vry la vrx  mov rez1, 0 ; salvam registrii  mov rez2, eax  mov rez3, edx  jmp ex  con2:  lea edx, mes5  call WriteString  mov eax, vry  cdq ; extindem  mov ebx, 4 ; inmultim vry \* 4  imul ebx  sub eax, vrx ; din rez - vrx  sbb edx, 0 ; adaugam CF  mov rez1, edx ; salvam registrii  mov rez2, eax  mov rez3, 0  jmp ex  ex:  lea edx, mes3  call WriteString ; afisarea mesajului mes3  call Crlf ; din rand nou  lea edx, mes6  call WriteString ; afisarea mesajului mes6  mov eax, rez1  call WriteInt  call Crlf ; din rand nou  lea edx, mes7  call WriteString  mov eax, rez2  call WriteInt ; afisarea rezultatului cu semn  call Crlf ; din rand nou  lea edx, mes8  call WriteString  mov eax, rez3  call WriteInt  call Crlf ; din rand nou  call Crlf  exit  main ENDP  END main |

Listingul programului.

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 04/08/19 22:27:18  programator.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  C .NOLIST  C .LIST  C  C INCLUDE VirtualKeys.inc  C ; VirtualKeys.inc  C .NOLIST  C .LIST  C  C  C .NOLIST  C .LIST  C  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  00000000 .data  00000000 20 56 61 6C 6F mes1 db " Valoarea X : ",0  61 72 65 61 20  58 20 3A 20 00  0000000F 20 56 61 6C 6F mes2 db " Valoarea Y : ",0  61 72 65 61 20  59 20 3A 20 00  0000001E 20 52 65 7A 75 mes3 db " Rezutatul este : ",0  74 61 74 75 6C  20 65 73 74 65  20 3A 20 00  00000031 20 43 6F 6E 64 mes4 db " Conditia 1 : ",0  69 74 69 61 20  31 20 3A 20 00  00000040 20 43 6F 6E 64 mes5 db " Conditia 2 : ",0  69 74 69 61 20  32 20 3A 20 00  0000004F 20 20 45 44 58 mes6 db " EDX = ",0  20 3D 20 00  00000058 20 20 45 41 58 mes7 db " EAX = ",0  20 3D 20 00  00000061 20 52 65 73 74 mes8 db " Rest = ",0  20 3D 20 00  0000006A 00000000 vrx dd 0  0000006E 00000000 vry dd 0  00000072 FFFFFFF1 vrz dd -15  00000076 00000000 rez1 dd 0  0000007A 00000000 rez2 dd 0  0000007E 00000000 rez3 dd 0  00000000 .code  00000000 main PROC  00000000 E8 00000000 E call Randomize  00000005 BA 00000000 R mov edx, OFFSET mes1 ; incarcam adresa mes1 in edx  0000000A E8 00000000 E call WriteString ; afisarea mes1  0000000F E8 00000000 E call ReadInt  00000014 A3 0000006A R mov vrx, eax ; salvarea valorii in variabila vrx  00000019 BA 0000000F R mov edx,OFFSET mes2  0000001E E8 00000000 E call WriteString ; afisarea mes2  00000023 E8 00000000 E call ReadInt  00000028 A3 0000006E R mov vry, eax ; salvarea valorii in variabila vry  0000002D A1 0000006A R mov eax, vrx ; incarcam x in eax  00000032 BB 00000002 mov ebx, 2 ; incarcam 2 in ebx  00000037 99 cdq  00000038 F7 FB idiv ebx ; impartim x la 2  0000003A 3B 05 0000006E R cmp eax, vry ; compararea x cu eax  00000040 7D 40 jge con2 ; daca >= cond 2    00000042 con1:  00000042 8D 15 00000031 R lea edx, mes4  00000048 E8 00000000 E call WriteString ; afisam mes4  0000004D E8 00000000 E call Crlf  00000052 BA 00000000 mov edx, 0  00000057 A1 0000006E R mov eax, vry ; incarcam vry in eax  0000005C 99 cdq ; extindem eax  0000005D 83 E8 38 sub eax, 56  00000060 73 01 jnc salt1  00000062 99 cdq ; extindem daca scaderea este cu imprumut  00000063 salt1:  00000063 8B 1D 00000072 R mov ebx, vrz ; incarcam -15  00000069 F7 FB idiv ebx ; impartim vry la vrx  0000006B C7 05 00000076 R mov rez1, 0 ; salvam registrii  00000075 A3 0000007A R mov rez2, eax  0000007A 89 15 0000007E R mov rez3, edx  00000080 EB 38 jmp ex  00000082 con2:  00000082 8D 15 00000040 R lea edx, mes5  00000088 E8 00000000 E call WriteString  0000008D A1 0000006E R mov eax, vry  00000092 99 cdq ; extindem  00000093 BB 00000004 mov ebx, 4 ; inmultim vry \* 4  00000098 F7 EB imul ebx  0000009A 2B 05 0000006A R sub eax, vrx ; din rez - vrx  000000A0 83 DA 00 sbb edx, 0 ; adaugam CF  000000A3 89 15 00000076 R mov rez1, edx ; salvam registrii  000000A9 A3 0000007A R mov rez2, eax  000000AE C7 05 0000007E R mov rez3, 0  00000000  000000B8 EB 00 jmp ex  000000BA ex:  000000BA 8D 15 0000001E R lea edx, mes3  000000C0 E8 00000000 E call WriteString ; afisarea mesajului mes3  000000C5 E8 00000000 E call Crlf ; din rand nou  000000CA 8D 15 0000004F R lea edx, mes6  000000D0 E8 00000000 E call WriteString ; afisarea mesajului mes6  000000D5 A1 00000076 R mov eax, rez1  000000DA E8 00000000 E call WriteInt  000000DF E8 00000000 E call Crlf ; din rand nou  000000E4 8D 15 00000058 R lea edx, mes7  000000EA E8 00000000 E call WriteString  000000EF A1 0000007A R mov eax, rez2  000000F4 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  000000F9 E8 00000000 E call Crlf ; din rand nou  000000FE 8D 15 00000061 R lea edx, mes8  00000104 E8 00000000 E call WriteString  00000109 A1 0000007E R mov eax, rez3  0000010E E8 00000000 E call WriteInt  00000113 E8 00000000 E call Crlf ; din rand nou  00000118 E8 00000000 E call Crlf  exit  0000011D 6A 00 \* push +000000000h  0000011F E8 00000000 E \* call ExitProcess  00000124 main ENDP  END main |

# Realizarea Conditiei 2 “Random32”

Codul programului pe 32 biți.

|  |
| --- |
| INCLUDE Irvine32.inc  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  .data  mes1 db "Valoarea X : ",0  mes2 db "Valoarea Y : ",0  mes3 db "Rezutatul este : ",0  mes4 db "Conditia 1 : ",0  mes5 db "Conditia 2 : ",0  mes6 db " EDX = ",0  mes7 db " EAX = ",0  mes8 db " Rest = ",0  vrx dd 0  vry dd 0  vrz dd -15  rez1 dd 0  rez2 dd 0  rez3 dd 0  .code  main PROC  call Randomize  mov edx, OFFSET mes1 ; incarcam adresa mes1 in edx  call WriteString ; afisarea mes1  call Random32  mov vrx, eax ; salvarea valorii in variabila vrx  call WriteInt ; afisarea rezultatului cu semn  call Crlf ; din rand nou  mov edx,OFFSET mes2  call WriteString ; afisarea mes2  call Random32  mov vry, eax ; salvarea valorii in variabila vry  call WriteInt ; afisarea rezultatului cu semn  call Crlf ; din rand nou  mov eax, vrx ; incarcam x in eax  mov ebx, 2 ; incarcam 2 in ebx  cdq  idiv ebx ; impartim x la 2  cmp eax, vry ; compararea x cu eax  jge con2 ; daca >= cond 2    con1:  lea edx, mes4  call WriteString ; afisam mes4  call Crlf  mov edx, 0  mov eax, vry ; incarcam vry in eax  cdq ; extindem eax  sub eax, 56  mov ebx, vrz ; incarcam -15  idiv ebx ; impartim vry la vrx  mov rez1, 0 ; salvam registrii  mov rez2, eax  mov rez3, edx  jmp ex  con2:  lea edx, mes5  call WriteString  mov eax, vry  cdq ; extindem  mov ebx, 4 ; inmultim vry \* 4  imul ebx  sub eax, vrx ; din rez - vrx  sbb edx, 0 ; adaugam CF  mov rez1, edx ; salvam registrii  mov rez2, eax  mov rez3, 0  jmp ex  ex:  lea edx, mes3  call WriteString ; afisarea mesajului mes3  call Crlf ; din rand nou  lea edx, mes6  call WriteString ; afisarea mesajului mes6  mov eax, rez1  call WriteInt  call Crlf ; din rand nou  lea edx, mes7  call WriteString  mov eax, rez2  call WriteInt ; afisarea rezultatului cu semn  call Crlf ; din rand nou  lea edx, mes8  call WriteString  mov eax, rez3  call WriteInt  call Crlf ; din rand nou  call Crlf  exit  main ENDP  END main |

Listingul programului pe 32 biți.

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 04/08/19 22:36:22  programator.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  C .NOLIST  C .LIST  C  C INCLUDE VirtualKeys.inc  C ; VirtualKeys.inc  C .NOLIST  C .LIST  C  C  C .NOLIST  C .LIST  C  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  00000000 .data  00000000 56 61 6C 6F 61 mes1 db "Valoarea X : ",0  72 65 61 20 58  20 3A 20 00  0000000E 56 61 6C 6F 61 mes2 db "Valoarea Y : ",0  72 65 61 20 59  20 3A 20 00  0000001C 52 65 7A 75 74 mes3 db "Rezutatul este : ",0  61 74 75 6C 20  65 73 74 65 20  3A 20 00  0000002E 43 6F 6E 64 69 mes4 db "Conditia 1 : ",0  74 69 61 20 31  20 3A 20 00  0000003C 43 6F 6E 64 69 mes5 db "Conditia 2 : ",0  74 69 61 20 32  20 3A 20 00  0000004A 20 20 45 44 58 mes6 db " EDX = ",0  20 3D 20 00  00000053 20 20 45 41 58 mes7 db " EAX = ",0  20 3D 20 00  0000005C 20 52 65 73 74 mes8 db " Rest = ",0  20 3D 20 00  00000065 00000000 vrx dd 0  00000069 00000000 vry dd 0  0000006D FFFFFFF1 vrz dd -15  00000071 00000000 rez1 dd 0  00000075 00000000 rez2 dd 0  00000079 00000000 rez3 dd 0  00000000 .code  00000000 main PROC  00000000 E8 00000000 E call Randomize  00000005 BA 00000000 R mov edx, OFFSET mes1 ; incarcam adresa mes1 in edx  0000000A E8 00000000 E call WriteString ; afisarea mes1  0000000F E8 00000000 E call Random32  00000014 A3 00000065 R mov vrx, eax ; salvarea valorii in variabila vrx  00000019 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  0000001E E8 00000000 E call Crlf ; din rand nou  00000023 BA 0000000E R mov edx,OFFSET mes2  00000028 E8 00000000 E call WriteString ; afisarea mes2  0000002D E8 00000000 E call Random32  00000032 A3 00000069 R mov vry, eax ; salvarea valorii in variabila vry  00000037 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  0000003C E8 00000000 E call Crlf ; din rand nou  00000041 A1 00000065 R mov eax, vrx ; incarcam x in eax  00000046 BB 00000002 mov ebx, 2 ; incarcam 2 in ebx  0000004B 99 cdq  0000004C F7 FB idiv ebx ; impartim x la 2  0000004E 3B 05 00000069 R cmp eax, vry ; compararea x cu eax  00000054 7D 3D jge con2 ; daca >= cond 2    00000056 con1:  00000056 8D 15 0000002E R lea edx, mes4  0000005C E8 00000000 E call WriteString ; afisam mes4  00000061 E8 00000000 E call Crlf  00000066 BA 00000000 mov edx, 0  0000006B A1 00000069 R mov eax, vry ; incarcam vry in eax  00000070 99 cdq ; extindem eax  00000071 83 E8 38 sub eax, 56  00000074 8B 1D 0000006D R mov ebx, vrz ; incarcam -15  0000007A F7 FB idiv ebx ; impartim vry la vrx  0000007C C7 05 00000071 R mov rez1, 0 ; salvam registrii  00000086 A3 00000075 R mov rez2, eax  0000008B 89 15 00000079 R mov rez3, edx  00000091 EB 38 jmp ex  00000093 con2:  00000093 8D 15 0000003C R lea edx, mes5  00000099 E8 00000000 E call WriteString  0000009E A1 00000069 R mov eax, vry  000000A3 99 cdq ; extindem  000000A4 BB 00000004 mov ebx, 4 ; inmultim vry \* 4  000000A9 F7 EB imul ebx  000000AB 2B 05 00000065 R sub eax, vrx ; din rez - vrx  000000B1 83 DA 00 sbb edx, 0 ; adaugam CF  000000B4 89 15 00000071 R mov rez1, edx ; salvam registrii  000000BA A3 00000075 R mov rez2, eax  000000BF C7 05 00000079 R mov rez3, 0  000000C9 EB 00 jmp ex  000000CB ex:  000000CB 8D 15 0000001C R lea edx, mes3  000000D1 E8 00000000 E call WriteString ; afisarea mesajului mes3  000000D6 E8 00000000 E call Crlf ; din rand nou  000000DB 8D 15 0000004A R lea edx, mes6  000000E1 E8 00000000 E call WriteString ; afisarea mesajului mes6  000000E6 A1 00000071 R mov eax, rez1  000000EB E8 00000000 E call WriteInt  000000F0 E8 00000000 E call Crlf ; din rand nou  000000F5 8D 15 00000053 R lea edx, mes7  000000FB E8 00000000 E call WriteString  00000100 A1 00000075 R mov eax, rez2  00000105 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  0000010A E8 00000000 E call Crlf ; din rand nou  0000010F 8D 15 0000005C R lea edx, mes8  00000115 E8 00000000 E call WriteString  0000011A A1 00000079 R mov eax, rez3  0000011F E8 00000000 E call WriteInt  00000124 E8 00000000 E call Crlf ; din rand nou  00000129 E8 00000000 E call Crlf  exit  0000012E 6A 00 \* push +000000000h  00000130 E8 00000000 E \* call ExitProcess  00000135 main ENDP  END main |

# Realizarea Conditiei 3 “RandomRange”

Codul programului pe 32 biți.

|  |
| --- |
| INCLUDE Irvine32.inc  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  .data  mes1 db " Valoarea X : ",0  mes2 db " Valoarea Y : ",0  mes3 db " Rezutatul este : ",0  mes4 db " Conditia 1 : ",0  mes5 db " Conditia 2 : ",0  mes6 db " EDX = ",0  mes7 db " EAX = ",0  mes8 db " Rest = ",0  vrx dd 0  vry dd 0  vrz dd -15  rez1 dd 0  rez2 dd 0  rez3 dd 0  .code  main PROC  call Randomize  mov edx, OFFSET mes1 ; incarcam adresa mes1 in edx  call WriteString ; afisarea mes1  mov eax, 100 ; range 0 - 100  call RandomRange  mov vrx, eax ; salvarea valorii in variabila vrx  call WriteInt ; afisarea rezultatului cu semn  call Crlf  mov edx,OFFSET mes2  call WriteString ; afisarea mes2  mov eax, 100 ; range 0 - 100  call RandomRange  mov vry, eax ; salvarea valorii in variabila vry  call WriteInt ; afisarea rezultatului cu semn  call Crlf  mov eax, vrx ; incarcam x in eax  mov ebx, 2 ; incarcam 2 in ebx  cdq  idiv ebx ; impartim x la 2  cmp eax, vry ; compararea x cu eax  jge con2 ; daca >= cond 2    con1:  lea edx, mes4  call WriteString ; afisam mes4  call Crlf  mov edx, 0  mov eax, vry ; incarcam vry in eax  cdq ; extindem eax  sub eax, 56  jnc salt1  cdq ; extindem daca scaderea este cu imprumut  salt1:  mov ebx, vrz ; incarcam -15  idiv ebx ; impartim vry la vrx  mov rez1, 0 ; salvam registrii  mov rez2, eax  mov rez3, edx  jmp ex  con2:  lea edx, mes5  call WriteString  mov eax, vry  cdq ; extindem  mov ebx, 4 ; inmultim vry \* 4  imul ebx  sub eax, vrx ; din rez - vrx  sbb edx, 0 ; adaugam CF  mov rez1, edx ; salvam registrii  mov rez2, eax  mov rez3, 0  jmp ex  ex:  lea edx, mes3  call WriteString ; afisarea mesajului mes3  call Crlf ; din rand nou  lea edx, mes6  call WriteString ; afisarea mesajului mes6  mov eax, rez1  call WriteInt  call Crlf ; din rand nou  lea edx, mes7  call WriteString  mov eax, rez2  call WriteInt ; afisarea rezultatului cu semn  call Crlf ; din rand nou  lea edx, mes8  call WriteString  mov eax, rez3  call WriteInt  call Crlf ; din rand nou  call Crlf  exit  main ENDP  END main |

Listingul programului pe 32 biți.

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 04/08/19 22:41:57  programator.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  ; z = { (y - 56)/(2 - 17), x < 2y } "Conditia 1"  ; { 4y - x, x/2 >= y } "Conditia 2"  00000000 .data  00000000 20 56 61 6C 6F mes1 db " Valoarea X : ",0  61 72 65 61 20  58 20 3A 20 00  0000000F 20 56 61 6C 6F mes2 db " Valoarea Y : ",0  61 72 65 61 20  59 20 3A 20 00  0000001E 20 52 65 7A 75 mes3 db " Rezutatul este : ",0  74 61 74 75 6C  20 65 73 74 65  20 3A 20 00  00000031 20 43 6F 6E 64 mes4 db " Conditia 1 : ",0  69 74 69 61 20  31 20 3A 20 00  00000040 20 43 6F 6E 64 mes5 db " Conditia 2 : ",0  69 74 69 61 20  32 20 3A 20 00  0000004F 20 20 45 44 58 mes6 db " EDX = ",0  20 3D 20 00  00000058 20 20 45 41 58 mes7 db " EAX = ",0  20 3D 20 00  00000061 20 52 65 73 74 mes8 db " Rest = ",0  20 3D 20 00  0000006A 00000000 vrx dd 0  0000006E 00000000 vry dd 0  00000072 FFFFFFF1 vrz dd -15  00000076 00000000 rez1 dd 0  0000007A 00000000 rez2 dd 0  0000007E 00000000 rez3 dd 0  00000000 .code  00000000 main PROC  00000000 E8 00000000 E call Randomize  00000005 BA 00000000 R mov edx, OFFSET mes1 ; incarcam adresa mes1 in edx  0000000A E8 00000000 E call WriteString ; afisarea mes1  0000000F B8 00000064 mov eax, 100 ; range 0 - 100  00000014 E8 00000000 E call RandomRange  00000019 A3 0000006A R mov vrx, eax ; salvarea valorii in variabila vrx  0000001E E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  00000023 E8 00000000 E call Crlf  00000028 BA 0000000F R mov edx,OFFSET mes2  0000002D E8 00000000 E call WriteString ; afisarea mes2  00000032 B8 00000064 mov eax, 100 ; range 0 - 100  00000037 E8 00000000 E call RandomRange  0000003C A3 0000006E R mov vry, eax ; salvarea valorii in variabila vry  00000041 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  00000046 E8 00000000 E call Crlf  0000004B A1 0000006A R mov eax, vrx ; incarcam x in eax  00000050 BB 00000002 mov ebx, 2 ; incarcam 2 in ebx  00000055 99 cdq  00000056 F7 FB idiv ebx ; impartim x la 2  00000058 3B 05 0000006E R cmp eax, vry ; compararea x cu eax  0000005E 7D 40 jge con2 ; daca >= cond 2    00000060 con1:  00000060 8D 15 00000031 R lea edx, mes4  00000066 E8 00000000 E call WriteString ; afisam mes4  0000006B E8 00000000 E call Crlf  00000070 BA 00000000 mov edx, 0  00000075 A1 0000006E R mov eax, vry ; incarcam vry in eax  0000007A 99 cdq ; extindem eax  0000007B 83 E8 38 sub eax, 56  0000007E 73 01 jnc salt1  00000080 99 cdq ; extindem daca scaderea este cu imprumut  00000081 salt1:  00000081 8B 1D 00000072 R mov ebx, vrz ; incarcam -15  00000087 F7 FB idiv ebx ; impartim vry la vrx  00000089 C7 05 00000076 R mov rez1, 0 ; salvam registrii  00000093 A3 0000007A R mov rez2, eax  00000098 89 15 0000007E R mov rez3, edx  0000009E EB 38 jmp ex  000000A0 con2:  000000A0 8D 15 00000040 R lea edx, mes5  000000A6 E8 00000000 E call WriteString  000000AB A1 0000006E R mov eax, vry  000000B0 99 cdq ; extindem  000000B1 BB 00000004 mov ebx, 4 ; inmultim vry \* 4  000000B6 F7 EB imul ebx  000000B8 2B 05 0000006A R sub eax, vrx ; din rez - vrx  000000BE 83 DA 00 sbb edx, 0 ; adaugam CF  000000C1 89 15 00000076 R mov rez1, edx ; salvam registrii  000000C7 A3 0000007A R mov rez2, eax  000000CC C7 05 0000007E R mov rez3, 0  00000000  000000D6 EB 00 jmp ex  000000D8 ex:  000000D8 8D 15 0000001E R lea edx, mes3  000000DE E8 00000000 E call WriteString ; afisarea mesajului mes3  000000E3 E8 00000000 E call Crlf ; din rand nou  000000E8 8D 15 0000004F R lea edx, mes6  000000EE E8 00000000 E call WriteString ; afisarea mesajului mes6  000000F3 A1 00000076 R mov eax, rez1  000000F8 E8 00000000 E call WriteInt  000000FD E8 00000000 E call Crlf ; din rand nou  00000102 8D 15 00000058 R lea edx, mes7  00000108 E8 00000000 E call WriteString  0000010D A1 0000007A R mov eax, rez2  00000112 E8 00000000 E call WriteInt ; afisarea rezultatului cu semn  00000117 E8 00000000 E call Crlf ; din rand nou  0000011C 8D 15 00000061 R lea edx, mes8  00000122 E8 00000000 E call WriteString  00000127 A1 0000007E R mov eax, rez3  0000012C E8 00000000 E call WriteInt  00000131 E8 00000000 E call Crlf ; din rand nou  00000136 E8 00000000 E call Crlf  exit  0000013B 6A 00 \* push +000000000h  0000013D E8 00000000 E \* call ExitProcess  00000142 main ENDP  END main |