**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Elaborare programelor în limbajul de asamblare**

**MASM în Visual Studio**

Lucrare de laborator nr.2

la disciplina Arhitectura Calculatoarelor

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

**Conducător: Colesnic Victor**

# Scopul lucrării

Se prezinta problemele principale legate de conversii de date, reprezentarea datelor întregi, reprezentarea întregilor in format BCD, reprezentarea caracterelor si a șirurilor de caractere, reprezentarea valorilor reale, elemente de memorie, tipuri de date utilizate si modurile de adresare a operanzilor.

# Realizarea pentru programul 3.1

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 03/05/19 13:32:00  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  00000000 .data  00000000 00000003 [ alfa DW 3 DUP(?)  0000  ]  00000000 .code  00000000 main proc  00000000 66| B8 0011 mov ax,17 ; adresare indirecta a operandului sursa  00000004 66| B8 0015 mov ax,10101b ;  00000008 66| B8 0003 mov ax,11b ;  0000000C 66| B8 0011 mov ax,21o ;  00000010 66| A3 mov alfa,ax ; Adresare directa a operandului destinatie  00000000 R    00000016 66| 8B C8 mov cx,ax ; Interschimba registrele ax si bx  00000019 66| 8B C3 mov ax,bx ; Folosind registrul cx  0000001C 66| 8B C1 mov ax,cx ;  0000001F 66| 93 xchg ax,bx ; Interschimba direct cele 2 registre.    00000021 66| BE 0002 mov si,2  00000025 66| 89 04 35 mov alfa[si],ax ; Adresare relativa cu registrul si  00000000 R  0000002D BE 00000002 mov esi,2  00000032 BB 00000000 R mov ebx, offset alfa ; Adresare imediata a operandului sursa  00000037 8D 1D 00000000 R lea ebx,alfa ; Acelasi efect  0000003D 8B 0C 33 mov ecx,[ebx][esi] ; Adresare bazata indexata a sursei  00000040 66| 8B 0D mov cx,alfa[2] ; Acelasi efect.  00000002 R  00000047 66| 8B 0D mov cx,[alfa+2] ; Acelasi efect  00000002 R  0000004E 66| BF 0004 mov di,4  00000052 C6 04 3B 55 mov byte ptr [ebx][edi],55h ;  00000056 BE 00000002 mov esi,2  0000005B BB 00000003 mov ebx,3  00000060 66| C7 84 33 mov alfa[ebx][esi],33h ; Adresare bazata indexata relativa a  00000000 R  0033  ; destinatiei  0000006A 66| C7 84 33 mov alfa[ebx+esi],33h ; Notatii echivalente  00000000 R  0033  00000074 66| C7 84 33 mov [alfa+ebx+esi],33h  00000000 R  0033  0000007E 66| C7 84 33 mov [ebx][esi]+alfa,33h  00000000 R  0033  exit  00000088 6A 00 \* push +000000000h  0000008A E8 00000000 E \* call ExitProcess  0000008F main ENDP  END main |

# Realizarea pentru programul 3.2

Listingul programului pe 32 biți.

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 03/19/19 14:13:22  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 ; 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  ; Sa se calculeze expresia aritmetica: e=((a+b\*c-d)/f+g\*h)/i  ; se considera a, d, f – cuvant b, c, g, h, i –byte  ; ca sa putem executa impartirea cu f convertim impartitorul la dublucuvant  ; ne vor interesa doar caturile impartirilor, rezultatul va fi de tip octet    00000000 .data  00000000 0005 a dw 5  00000002 06 b db 6  00000003 0A cd db 10  00000004 0005 d dw 5  00000006 0006 f dw 6  00000008 0A g db 10  00000009 0B h db 11  0000000A 0A i db 10  0000000B 0000 interm dw ?  0000000D 00 rez db ?  00000000 .code  00000000 main proc  00000000 B8 00000000 mov eax,0  00000005 A0 00000002 R mov al, b  0000000A F6 2D 00000003 R imul cd ; in ax avem b\*c  00000010 66| 03 05 add ax, a ; ax=b\*c+a  00000000 R  00000017 66| 2B 05 sub ax, d ; ax=b\*c+a-d  00000004 R  0000001E 66| 99 cwd ; am convertit cuvantul in dublu cuvantul  00000020 66| F7 3D idiv f ; obtinem catul in ax si restul în dx ax=(a+b\*c-d)/f  00000006 R  00000027 66| A3 mov interm, ax ; interm=(a+b\*c-d)/f  0000000B R  0000002D A0 00000008 R mov al, g  00000032 F6 2D 00000009 R imul h ; ax=g\*h  00000038 66| 03 05 add ax, interm ; ax=(a+b\*c-d)/f+g\*h  0000000B R  0000003F F6 3D 0000000A R idiv i ; se obtine catul in al si restul in ah  00000045 A2 0000000D R mov rez, al  exit  0000004A 6A 00 \* push +000000000h  0000004C E8 00000000 E \* call ExitProcess  00000051 main ENDP  END main  ; Date de test : vom obţine rezultatul ((a+b\*c-d)/f+g\*h)/i=((5+6\*10-5)/6+10\*11)/10= 12  \_Microsoft (R) Macro Assembler Version 14.16.27026.1 03/19/19 14:13:22  programator.asm Symbols 2 - 1 |

# Realizarea pentru programul 4

Varianta 9.

Exemplu :

Codul programului 4

|  |
| --- |
| INCLUDE Irvine32.inc  ; z=(((a+1)\*(a+1)+2)^2)/(b\*b+c\*c)  ; z=((9+1)\*(9+1)+2)\*2/(6\*6+7\*7)  ; z=( 10 \* 10 +2)\*2/(36 + 49)  ; z=( 102 )\*2/( 85 )  ; z=( 204 ) / 85  ; z= 170 / 85 = 2 si 204 - 170 = 34    .data  a db 9  b db 6  cc db 7  rez dw ?  .code  main proc  mov eax,0  mov al, a ; a  add ax, 1 ; a + 1    imul ax, ax ; (a+1) \* (a+1)  add ax, 2 ; (a+1)\*(a+1) + 2  imul ax, ax ; ((a+1)\*(a+1)+2) ^ 2    mov rez, ax ; rez = r  mov ax, 0 ; zeroficam    mov al, b ; b  imul ax, ax ; b\*b  mov bx, ax ; salvez prod din AX in BX  mov ax, 0    mov al, cc ; c  imul ax, ax ; c\*c  add ax, bx ; (b\*b) + (c\*c)    mov bx, ax ; salvez a doua suma din AX in BX  mov ax, rez ; incarc prima suma din Rez in AX  div bl ; AX / bl = AL , AH - restul  mov bl, ah ; salvam restul  mov ah, 0 ; separam AL de AH  call WriteInt ; afisam AX la ecram  call Crlf ; rind nou  mov al, bl ; incarcam citul din BL in AL  call WriteInt ; afisam AX  call Crlf ; rind nou  exit  main ENDP  END main |

Listingul programului pe 32 biți.

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.16.27026.1 03/19/19 15:26:30  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=(((a+1)\*(a+1)+2)^2)/(b\*b+c\*c)  ; z=((9+1)\*(9+1)+2)\*2/(6\*6+7\*7)  ; z=( 10 \* 10 +2)\*2/(36 + 49)  ; z=( 102 )\*2/( 85 )  ; z=( 204 ) / 85  ; z= 170 / 85 = 2 si 204 - 170 = 34    00000000 .data  00000000 09 a db 9  00000001 06 b db 6  00000002 07 cc db 7  00000003 0000 rez dw ?  00000000 .code  00000000 main proc  00000000 B8 00000000 mov eax,0  00000005 A0 00000000 R mov al, a ; a  0000000A 66| 83 C0 01 add ax, 1 ; a + 1    0000000E 66| 0F AF C0 imul ax, ax ; (a+1) \* (a+1)  00000012 66| 83 C0 02 add ax, 2 ; (a+1)\*(a+1) + 2  00000016 66| 0F AF C0 imul ax, ax ; ((a+1)\*(a+1)+2) ^ 2    0000001A 66| A3 mov rez, ax ; rez = r  00000003 R  00000020 66| B8 0000 mov ax, 0 ; zeroficam    00000024 A0 00000001 R mov al, b ; b  00000029 66| 0F AF C0 imul ax, ax ; b\*b  0000002D 66| 8B D8 mov bx, ax ; salvez prod din AX in BX  00000030 66| B8 0000 mov ax, 0    00000034 A0 00000002 R mov al, cc ; c  00000039 66| 0F AF C0 imul ax, ax ; c\*c  0000003D 66| 03 C3 add ax, bx ; (b\*b) + (c\*c)    00000040 66| 8B D8 mov bx, ax ; salvez a doua suma din AX in BX  00000043 66| A1 mov ax, rez ; incarc prima suma din Rez in AX  00000003 R  00000049 F6 F3 div bl ; AX / bl = AL , AH - restul  0000004B 8A DC mov bl, ah ; salvam restul  0000004D B4 00 mov ah, 0 ; separam AL de AH  0000004F E8 00000000 E call WriteInt ; afisam AX la ecram  00000054 E8 00000000 E call Crlf ; rind nou  00000059 8A C3 mov al, bl ; incarcam citul din BL in AL  0000005B E8 00000000 E call WriteInt ; afisam AX  00000060 E8 00000000 E call Crlf ; rind nou  exit  0000006B 6A 00 \* push +000000000h  0000006D E8 00000000 E \* call ExitProcess  00000072 main ENDP  END main  \_Microsoft (R) Macro Assembler Version 14.16.27026.1 03/19/19 15:26:30  programator.asm Symbols 2 - 1 |