**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 1

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

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

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

**Chișinău 2024**

# Tema lucrării:

Elaborarea programelor în limbajul de asamblare MASM în Visual Studio

# Scopul lucrării:

Lucrarea urmărește familiarizarea studenților cu regiştrii microprocesoarelor pe 16, 32 şi 64

de biți, cu mediul de dezvoltare a programelor Visual Studio. Se prezintă setările mediului Visual

Studio necesare pentru a elabora, rula şi depăna aplicaţiile elaborate în limbajul de asamblare MASM

utilizând regiştri pe 16, 32 şi 64 de biţi

# Codurile sursă:

Programul pe 32 biți:

|  |
| --- |
| .model flat, stdcall  .386  INCLUDE Irvine32.inc  .data  Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]',0  Dad DB 13,10,'Vei deveni!',13,10,0  Nud DB 13,10,'Vei deveni filosof!',13,10,0  .code  main PROC  mov edx,OFFSET Promt  call WriteString ;cheama functii din biblioteca Irvine  call ReadChar  cmp al,'y' ;compara daca caracterul introdus este y  jz IsDad ;va afisa sirul Dad  cmp al,'n' ;compara daca caracterul introdus este n  jz IsNud ;va afisa sirul Nud  IsDad: mov edx,OFFSET Dad  call WriteString  jmp ex  IsNud: mov edx,OFFSET Nud  call WriteString  ex:  exit  main ENDP  END main |

Programul pe 16 biți:

|  |
| --- |
| INCLUDE Irvine16.inc  .DATA  Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]$'  Dad DB 13,10,'Vei deveni!',13,10,'$'  Nud DB 13,10,'Vei deveni filosof!',13,10,'$'  .CODE  main PROC  mov ax,@data  mov ds,ax  mov dx,OFFSET Promt  mov ah,9  Int 21h  mov ah,1  Int 21h  cmp al,'y'  jz IsDad  cmp al,'n'  jz IsNud  IsDad: mov dx,OFFSET Dad  Jmp SHORT Disp  IsNud: mov dx,OFFSET Nud  Disp: mov ah,9  Int 21h  mov ah,1  Int 21h  Exit  main ENDP  END main |

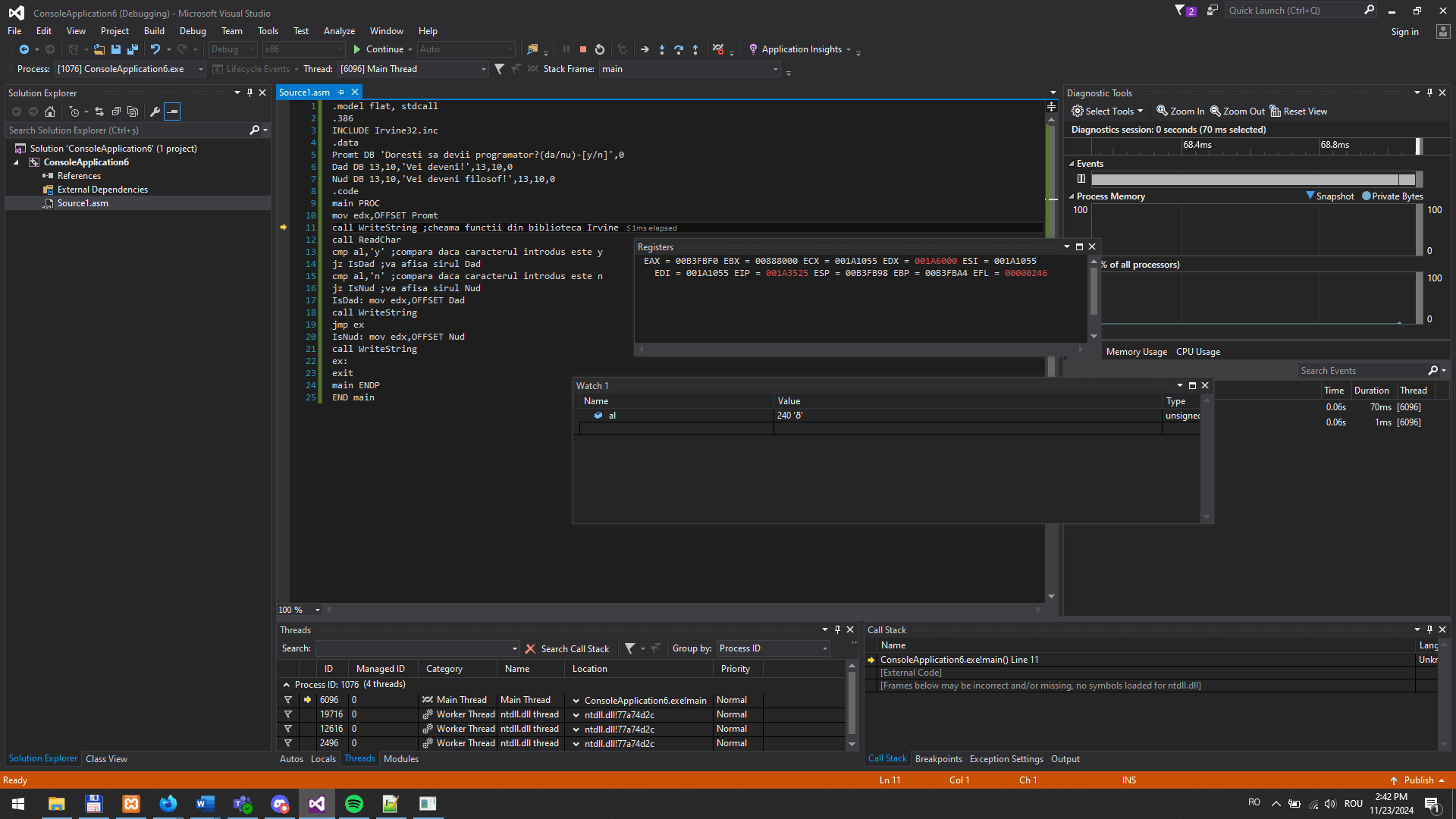
# Fișierele listing:

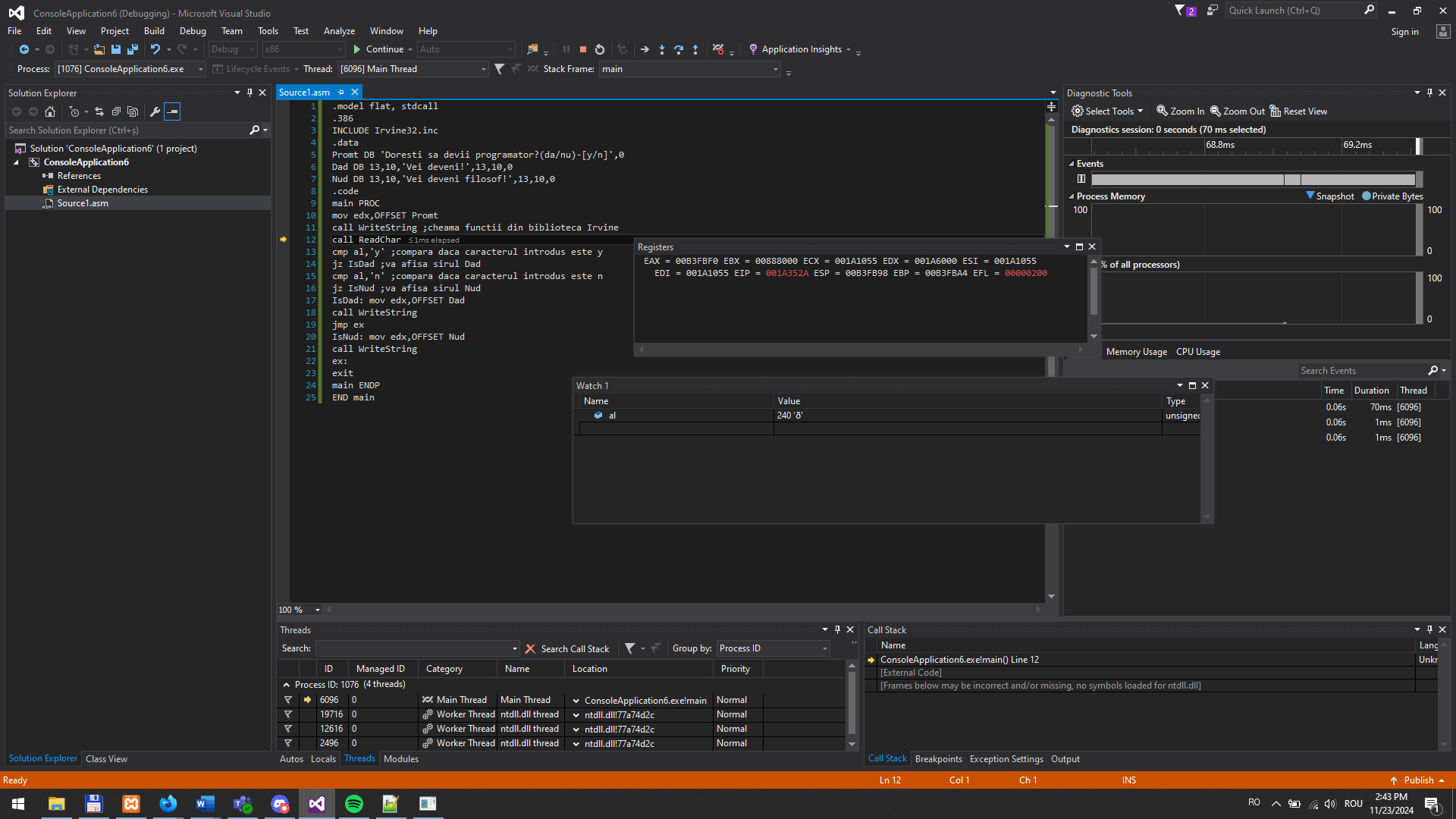
Listing-ul pentru 32 biți:

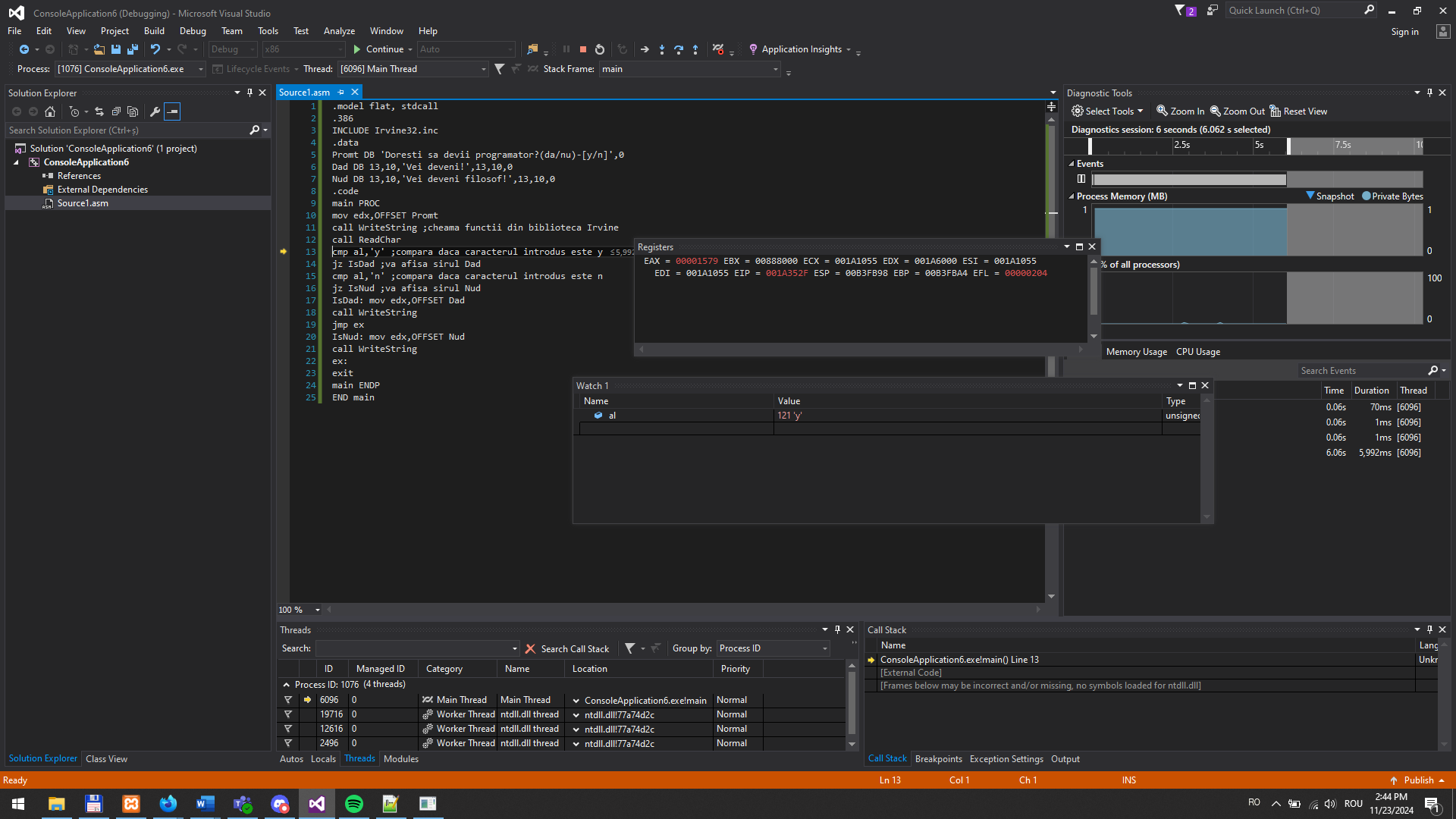
|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 14:23:07  Source1.asm Page 1 - 1  .model flat, stdcall  .386  INCLUDE Irvine32.inc  C ; Include file for Irvine32.lib (Irvine32.inc)  C  C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive  C  C INCLUDE SmallWin.inc ; 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 44 6F 72 65 73 Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]',0  74 69 20 73 61  20 64 65 76 69  69 20 70 72 6F  67 72 61 6D 61  74 6F 72 3F 28  64 61 2F 6E 75  29 2D 5B 79 2F  6E 5D 00  0000002B 0D 0A 56 65 69 Dad DB 13,10,'Vei deveni!',13,10,0  20 64 65 76 65  6E 69 21 0D 0A  00  0000003B 0D 0A 56 65 69 Nud DB 13,10,'Vei deveni filosof!',13,10,0  20 64 65 76 65  6E 69 20 66 69  6C 6F 73 6F 66  21 0D 0A 00  00000000 .code  00000000 main PROC  00000000 BA 00000000 R mov edx,OFFSET Promt  00000005 E8 00000000 E call WriteString  0000000A E8 00000000 E call ReadChar  0000000F 3C 79 cmp al,'y'  00000011 74 04 jz IsDad  00000013 3C 6E cmp al,'n'  00000015 74 0C jz IsNud  00000017 BA 0000002B R IsDad: mov edx,OFFSET Dad  0000001C E8 00000000 E call WriteString  00000021 EB 0A jmp ex  00000023 BA 0000003B R IsNud: mov edx,OFFSET Nud  00000028 E8 00000000 E call WriteString  0000002D ex:  exit  00000034 main ENDP  END main  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 14:23:07  Source1.asm Symbols 2 - 1 |

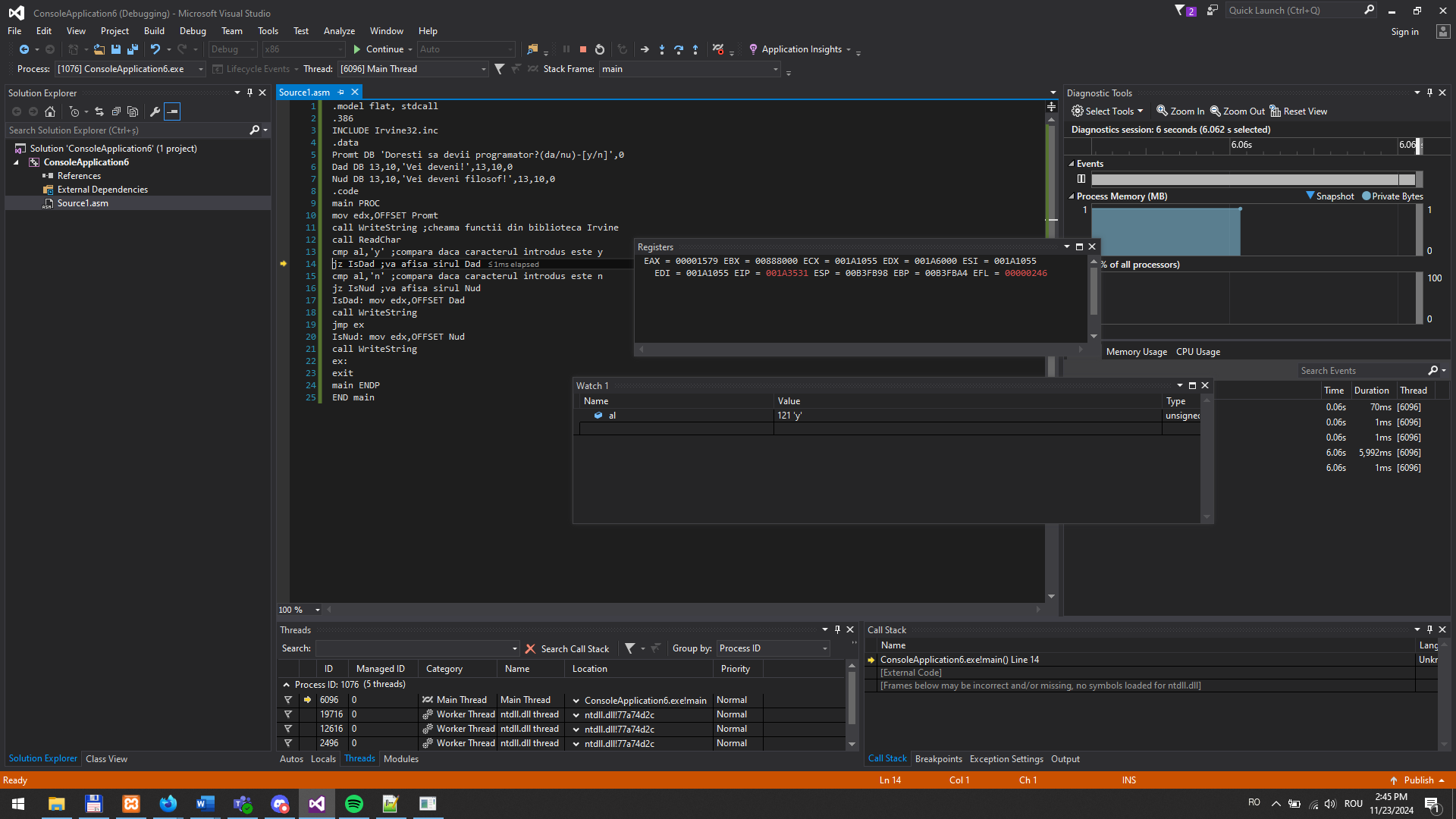
Listing-ul programului pe 16-biți:

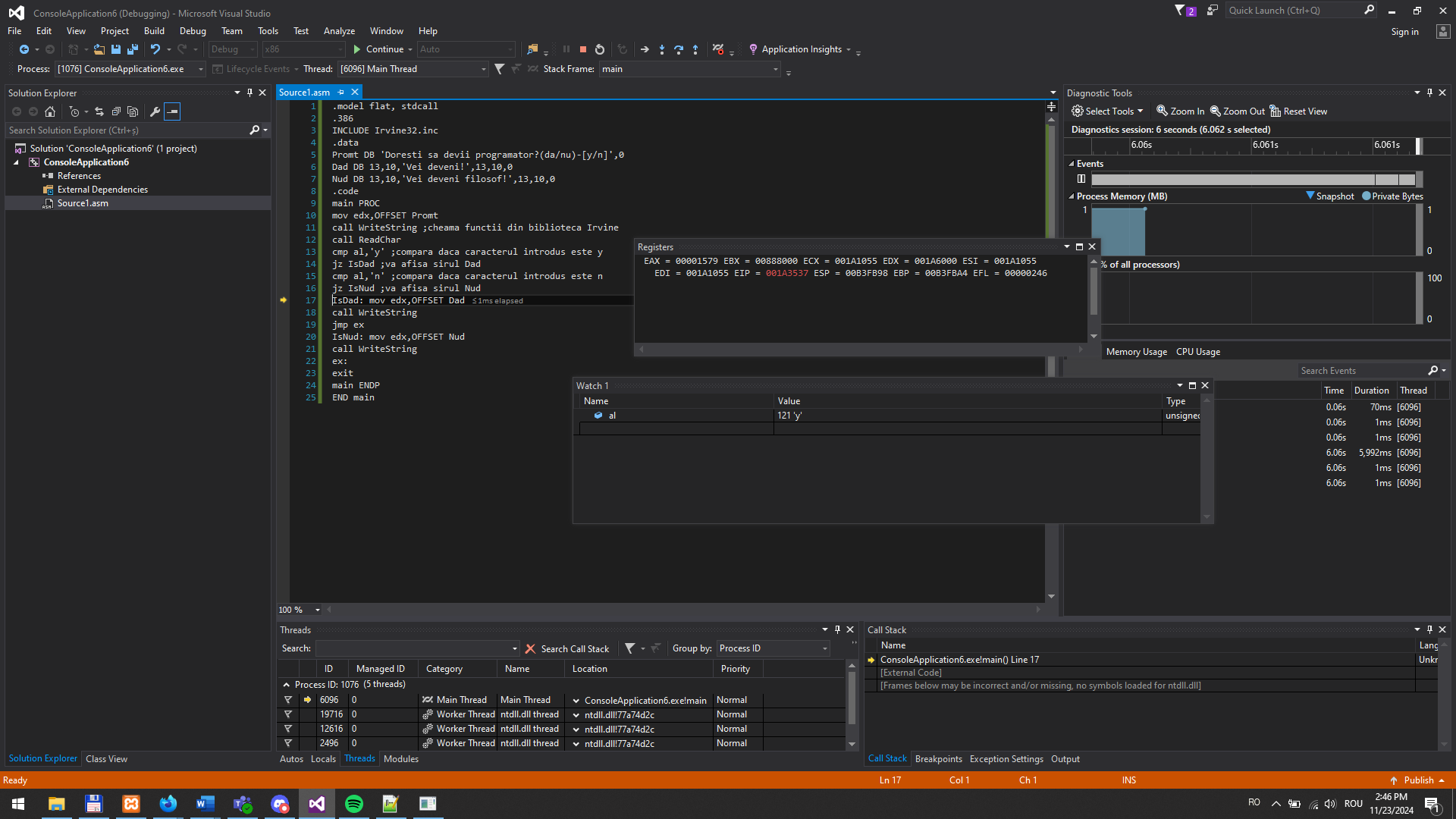
|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 15:50:09  program1.asm Page 1 - 1  INCLUDE Irvine16.inc  C ; Irvine16.inc - Include file for programs using  C ; the Irvine16.lib (Real-address mode library).  C  C ; Last update: 7/29/05  C  C .NOLIST  C .LIST  C  0000 .DATA  0000 44 6F 72 65 73 74 Promt DB 'Doresti sa devii programator?(da/nu)-[y/n]$'  69 20 73 61 20 64  65 76 69 69 20 70  72 6F 67 72 61 6D  61 74 6F 72 3F 28  64 61 2F 6E 75 29  2D 5B 79 2F 6E 5D  24  002B 0D 0A 56 65 69 20 Dad DB 13,10,'Vei deveni!',13,10,'$'  64 65 76 65 6E 69  21 0D 0A 24  003B 0D 0A 56 65 69 20 Nud DB 13,10,'Vei deveni filosof!',13,10,'$'  64 65 76 65 6E 69  20 66 69 6C 6F 73  6F 66 21 0D 0A 24  0000 .CODE  0000 main PROC  0000 B8 ---- R mov ax,@data  0003 8E D8 mov ds,ax  0005 BA 0000 R mov dx,OFFSET Promt  0008 B4 09 mov ah,9  000A CD 21 Int 21h  000C B4 01 mov ah,1  000E CD 21 Int 21h  0010 3C 79 cmp al,'y'  0012 74 04 jz IsDad  0014 3C 6E cmp al,'n'  0016 74 05 jz IsNud  0018 BA 002B R IsDad: mov dx,OFFSET Dad  001B EB 03 Jmp SHORT Disp  001D BA 003B R IsNud: mov dx,OFFSET Nud  0020 B4 09 Disp: mov ah,9  0022 CD 21 Int 21h  0024 B4 01 mov ah,1  0026 CD 21 Int 21h  Exit  002C main ENDP  END main  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/23/24 15:50:09  program1.asm Symbols 2 - 1 |

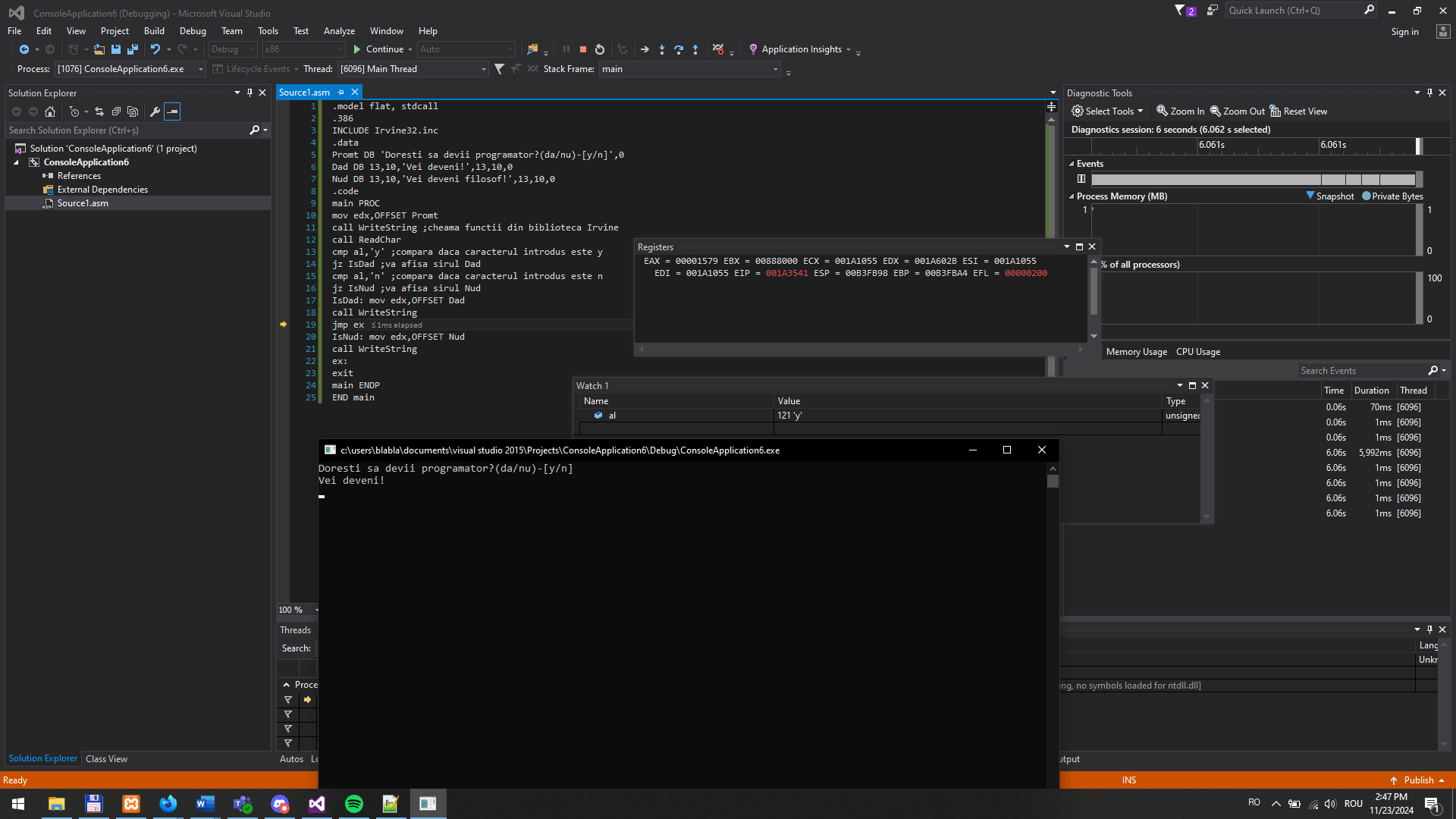




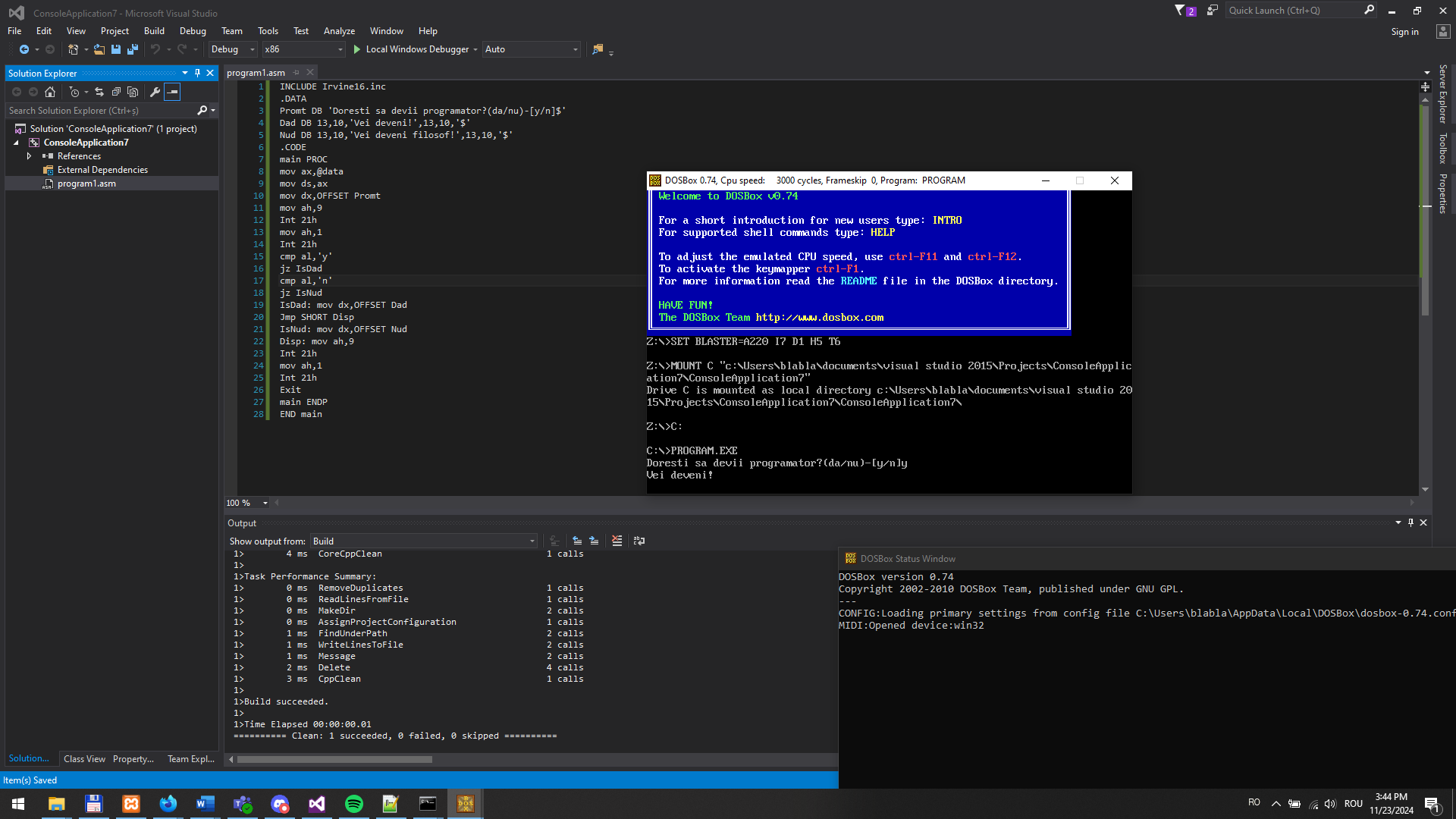








Rezultatul executării programului pe 16 biți:



# Concluzii:

În urma efectuării lucrării de laborator, am obținut cunoștințe în rularea programelor assembly atât pe 32 biți cât și pe 16 biți. Astfel, am remarcat că rularea celor pe 32 biți este mai simplă, iar procedurile sunt mai puțin complexe decât la celalt, luând în considerare că accesarea și manipularea memoriei este îmbunătățită. De asemenea, am observant că programele pe 16 biți nu pot fi rulate în mod normal, fiind nevoie de virtual box, deoarece accesarea blocurilor de memorie nu poate fi efectuată de procesoarele moderne. Din această cauză, depanarea programului pe 16 biți nu este posibilă în Visual Studio, iar executarea lui a fost efectuată prin intermediul a DOSBox, care reprezintă mediul potrivit pentru interpretarea comenzilor de 16 biți.