**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 4

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

***Varianta 3***

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

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

**Chișinău 2024**

# Scopul lucrării:

Lucrarea urmărește deprinderea studenților cu proiectarea și implementarea programelor cu subprograme și macroinstrucțiuni, cu ramificații și bucle in limbaj de asamblare. Se prezintă câteva exemple tipice de astfel de programe, incluzând câteva metode elementare de sortare şi căutare. De asemenea se prezintă câteva implementări de operații aritmetice care necesită utilizarea unor structuri de control de tip ramificare şi buclare. Se vor utiliza instrucțiunile de comparare, salt şi buclare. Se prezintă de asemenea şi câteva exemple de utilizare a instrucțiunilor logice.

# Sarcina lucrării:

Elaboraţi un program care convertește un număr de 16 cifre codificate BCD împachetat începând de la adresa packed într‑un număr BCD neîmpachetat începând de la adresa unpacked. Modificați apoi secvența anterioară astfel încât să se obțină numărul codificat sub forma unui şir de caractere ASCII şi să se afișeze apoi șirul respectiv.

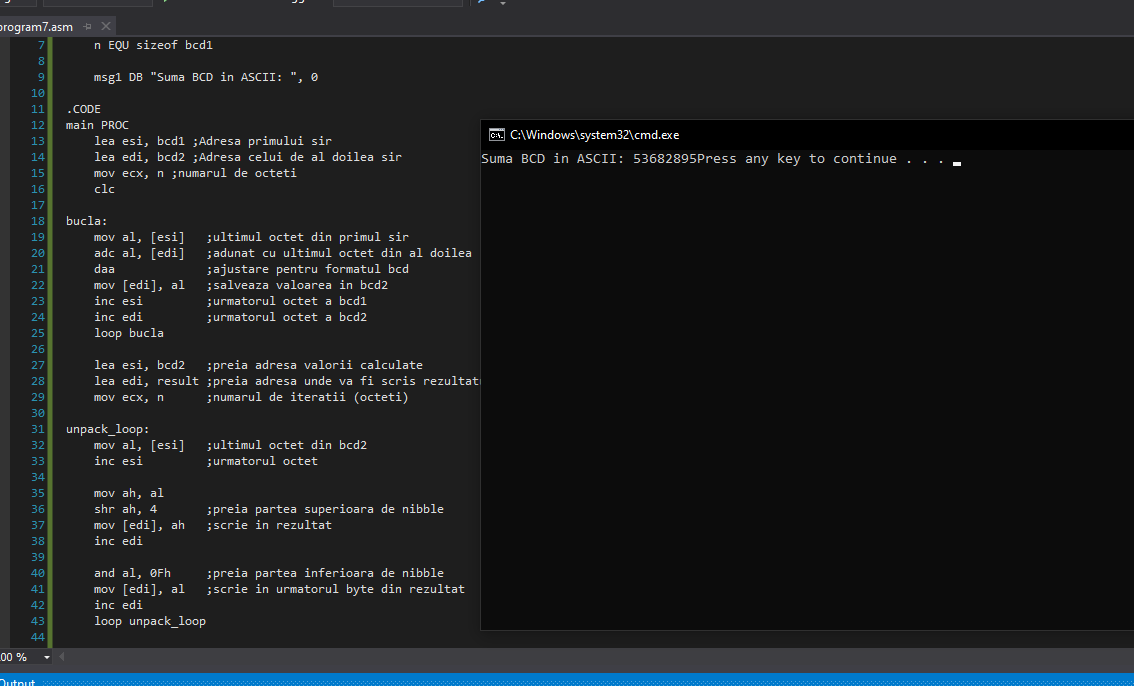
# Codul sursă:

|  |
| --- |
| INCLUDE Irvine32.inc  .DATA  bcd1 DB 22h,41h,75h,32h  bcd2 DB 31h,27h,53h,62h  result DB 8 DUP(0)  asciiStr DB 9 DUP(0), '$'  n EQU sizeof bcd1  msg1 DB "Suma BCD in ASCII: ", 0  .CODE  main PROC  lea esi, bcd1 ;Adresa primului sir  lea edi, bcd2 ;Adresa celui de al doilea sir  mov ecx, n ;numarul de octeti  clc  bucla:  mov al, [esi] ;ultimul octet din primul sir  adc al, [edi] ;adunat cu ultimul octet din al doilea  daa ;ajustare pentru formatul bcd  mov [edi], al ;salveaza valoarea in bcd2  inc esi ;urmatorul octet a bcd1  inc edi ;urmatorul octet a bcd2  loop bucla  lea esi, bcd2 ;preia adresa valorii calculate  lea edi, result ;preia adresa unde va fi scris rezultatul  mov ecx, n ;numarul de iteratii (octeti)  unpack\_loop:  mov al, [esi] ;ultimul octet din bcd2  inc esi ;urmatorul octet  mov ah, al  shr ah, 4 ;preia partea superioara de nibble  mov [edi], ah ;scrie in rezultat  inc edi  and al, 0Fh ;preia partea inferioara de nibble  mov [edi], al ;scrie in urmatorul byte din rezultat  inc edi  loop unpack\_loop  lea esi, result ;preia adresa rezultatului  lea edi, asciiStr ;adresa unde va scrie sirul in ascii  mov ecx, 8 ;numarul de iteratii  ascii\_conversion:  mov al, [esi]  add al, 30h ;conversia in ascii (adunarea cu caracterul 0)  mov [edi], al  inc esi  inc edi  loop ascii\_conversion  lea edx, msg1  call WriteString  lea edx, asciiStr  call WriteString  exit  main ENDP  END main |

# Fișierele listing:

|  |
| --- |
| Microsoft (R) Macro Assembler Version 14.00.24210.0 11/24/24 19:41:45  program7.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 22 41 75 32 bcd1 DB 22h,41h,75h,32h  00000004 31 27 53 62 bcd2 DB 31h,27h,53h,62h  00000008 00000008 [ result DB 8 DUP(0)  00  ]  00000010 00000009 [ asciiStr DB 9 DUP(0), '$'  00  ] 24  = 00000004 n EQU sizeof bcd1  0000001A 53 75 6D 61 20 msg1 DB "Suma BCD in ASCII: ", 0  42 43 44 20 69  6E 20 41 53 43  49 49 3A 20 00  00000000 .CODE  00000000 main PROC  00000000 8D 35 00000000 R lea esi, bcd1 ;Adresa primului sir  00000006 8D 3D 00000004 R lea edi, bcd2 ;Adresa celui de al doilea sir  0000000C B9 00000004 mov ecx, n ;numarul de octeti  00000011 F8 clc  00000012 bucla:  00000012 8A 06 mov al, [esi] ;ultimul octet din primul sir  00000014 12 07 adc al, [edi] ;adunat cu ultimul octet din al doilea  00000016 27 daa ;ajustare pentru formatul bcd  00000017 88 07 mov [edi], al ;salveaza valoarea in bcd2  00000019 46 inc esi ;urmatorul octet a bcd1  0000001A 47 inc edi ;urmatorul octet a bcd2  0000001B E2 F5 loop bucla  0000001D 8D 35 00000004 R lea esi, bcd2 ;preia adresa valorii calculate  00000023 8D 3D 00000008 R lea edi, result ;preia adresa unde va fi scris rezultatul  00000029 B9 00000004 mov ecx, n ;numarul de iteratii (octeti)  0000002E unpack\_loop:  0000002E 8A 06 mov al, [esi] ;ultimul octet din bcd2  00000030 46 inc esi ;urmatorul octet  00000031 8A E0 mov ah, al  00000033 C0 EC 04 shr ah, 4 ;preia partea superioara de nibble  00000036 88 27 mov [edi], ah ;scrie in rezultat  00000038 47 inc edi  00000039 24 0F and al, 0Fh ;preia partea inferioara de nibble  0000003B 88 07 mov [edi], al ;scrie in urmatorul byte din rezultat  0000003D 47 inc edi  0000003E E2 EE loop unpack\_loop  00000040 8D 35 00000008 R lea esi, result ;preia adresa rezultatului  00000046 8D 3D 00000010 R lea edi, asciiStr ;adresa unde va scrie sirul in ascii  0000004C B9 00000008 mov ecx, 8 ;numarul de iteratii  00000051 ascii\_conversion:  00000051 8A 06 mov al, [esi]  00000053 04 30 add al, 30h ;conversia in ascii (adunarea cu caracterul 0)  00000055 88 07 mov [edi], al  00000057 46 inc esi  00000058 47 inc edi  00000059 E2 F6 loop ascii\_conversion  0000005B 8D 15 0000001A R lea edx, msg1  00000061 E8 00000000 E call WriteString  00000066 8D 15 00000010 R lea edx, asciiStr  0000006C E8 00000000 E call WriteString  exit  00000078 main ENDP  END main  Microsoft (R) Macro Assembler Version 14.00.24210.0 11/24/24 19:41:45  program7.asm Symbols 2 - 1 |

Rezultatul executării programului:



# Concluzii:

În urma efectuării lucrării de laborator, am implementat un program în limbajul assembly care calculează suma a două numere reprezentate în format BCD (Binary-Coded Decimal) utilizând operații avansate de procesare a datelor și proceduri din biblioteca Irvine32. Programul demonstrează utilizarea eficientă a registrelor, a instrucțiunilor aritmetice (ADC, DAA) și a operațiilor logice pentru a transforma rezultatele în format neîmpachetat și ulterior în șiruri de caractere ASCII pentru afișare. Conversia automată a rezultatelor din BCD în format ASCII și afișarea acestora pe consolă a fost realizată utilizând funcții moderne de I/O din Irvine32, eliminând necesitatea apelurilor de tip DOS. De asemenea, programul a fost testat pe un set fix de date pentru a verifica funcționalitatea fiecărei etape: adunarea BCD împachetată, deîmpachetarea rezultatelor și generarea șirului ASCII.