****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software i Automatică**

**Raport**

**Lucrarea de laborator №5**

Disciplina: Arhitectura Calculatoarelor.

Tema: PRELUCRAREA SIRURILOR

UTILIZAREA TABELELOR PENTRU CONVERSII DE CODURI

**Realizat:** Bunescu Gabriel, TI-207

**Profesor:** asist.univ.mag.Secrieru Adrian

Chișinău 2022

**Scopul lucrării:**

Se prezinta probleme legate de prelucrarea in limbaj de asamblare a șirurilor. Pentru aceasta se recomanda utilizarea instrucțiunilor speciale pentru tratarea șirurilor. Se prezinta de asemenea tehnici specifice limbajului de asamblare pentru realizarea conversiilor de coduri, bazate pe utilizarea tabelelor de conversie.

**Mersul lucrării:**

**Exemplul 1:**

INCLUDE Irvine32.inc

.DATA

sir1 DB 100 DUP(7)

sir2 DB 100 DUP(?)

.CODE

main proc

mov esi,OFFSET sir1

mov edi,OFFSET sir2

mov ecx,LENGTHOF sir1

muta: mov al,[esi]

mov [edi],al

inc esi

inc edi

loop muta

exit

main ENDP

END main

**Codul sursă:**

include Irvine32.inc

.data

str1Mes db "Str1: ",0

str2Mes db "Str2: ",0

resultMes db "Result: ",0

str1MaxLen = 100

str2MaxLen = 100

str1 byte str1MaxLen+1 DUP(?)

str2 byte str2MaxLen+1 DUP(?)

str3 byte str1MaxLen+str2MaxLen+1 DUP(?)

str1Len dd ?

str2Len dd ?

str3Len dd ?

concat MACRO src1, src2, dest, src1Len, src2Len

cld

mov esi, offset src1

mov edi, offset dest

mov ecx, src1Len

rep movsb

mov esi, offset src2

mov edi, offset dest

add edi, src1Len

mov ecx, src2Len

rep movsb

replaceChar dest

ENDM

replaceChar MACRO str

LOCAL nextChar, nd

mov edi, offset str

dec edi

nextChar:

inc edi

mov ebx, [edi]

cmp bl, 0

je nd

cmp bl, 32 ; 32 = ' '

jne nextChar

mov ebx, 0

mov bl, 64 ; 64 = '@'

mov [edi], bl

jmp nextChar

nd:

ENDM

inputValues MACRO

mov edx, offset str1Mes

call WriteString

mov edx, offset str1

mov ecx, str1MaxLen

call ReadString

mov str1Len, eax

mov edx, offset str2Mes

call WriteString

mov edx, offset str2

mov ecx, str2MaxLen

call ReadString

mov str2Len, eax

add eax, str1Len

mov str3Len, eax

ENDM

outputResults MACRO

mov al, 10

call WriteChar

mov edx, offset resultMes

call WriteString

mov edx, offset str3

call WriteString

mov al, 10

call WriteChar

ENDM

.code

main proc

inputValues

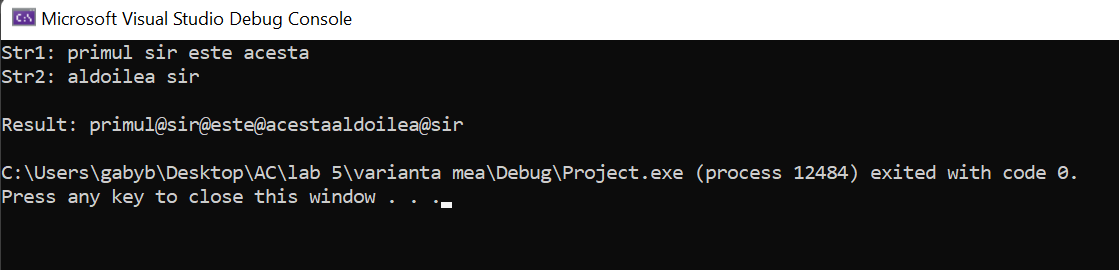
concat str1, str2, str3, str1Len, str2Len

outputResults

exit

main endp

end main



**Concluzie:**

În urma realizării lucrării de laborator nr.5, am studiat mai detaliat instrucțiunile MOVSB, CMPSB, SCASB, STOSB si LODSB. Aceste cunoștințe mi-au permis sa efectuez lucrarea de laborator unde am creat un program ce concateneaza două șiruri, introduse de utilizator, salveaza răspunsul în alt sir și inlocuieste fiecare caracter al liber spațiului cu caracterul ‘@’.