МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

**Лабораторна робота №4**

**Організація взаємозв'язку програм мовою Асемблера з С++ програмами**

**Варіант №21**

Виконав:

Студент групи КВ-51

Тимошенко І.О.

Київ

2016

**Постановка задачі**

Розробити функцію **bool Biggr(byte\* M1, byte\* M2, short len)**, де M1, M2 – надвеликі

цілі додатні числа (байтові масиви довжиною len). Операція – якщо М1 > М2, то

значення Biggr – True, інакше – False. Повинні використовуватись команди для 32-

розрядних даних. Якщо значення len не кратно 4, то при необхідності для порівняння

останніх байт використовувати команди для 8-розрядних даних.

**Код програми**

**Laba4.cpp**

#include <stdio.h>

#define n 62 // кількість байтів у надвеликому числі

typedef unsigned char byte; // для роботи з байтами використовується тип char

extern "C"

{

void BigShowN(byte\* p1, int p2);

bool Biggr(byte \*M1, byte \*M2, short len);

}

int main()

{

byte x[n], y[n]; //надвеликі числа

for (int i = 0; i<n; i++)

{

x[i] = i;

y[i] = n - i;

}

printf("x:");

BigShowN(x, n);

printf("y:");

BigShowN(y, n);

if (Biggr(x, y, n))

printf("x > y: true\n");

else

printf("x > y: false\n");

return 0;

}

**Biggr.asm**

.686

.model flat,C

public Biggr

.code

Biggr proc

push ebp

mov ebp,esp

mov esi, [ebp + 8]

mov edi, [ebp + 12]

mov ecx, [ebp + 16]

test ecx, 00000011b

jz dw\_loop

b\_loop:

dec ecx

mov al, byte ptr [esi + ecx]

mov bl, byte ptr [edi + ecx]

test ecx, 00000011b

jz dw\_loop

cmp al, bl

je b\_loop

jmp loop\_end

dw\_loop:

sub ecx, 4

mov eax, dword ptr [esi + ecx]

mov ebx, dword ptr [edi + ecx]

cmp ecx, 0

jl false

cmp al, bl

je dw\_loop

loop\_end:

jbe false

mov al, 1

jmp if\_end

false:

mov ax, 0

if\_end:

pop ebp

ret

Biggr endp

End