Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 10

З дисципліни «Системне програмування»

ВИКОНАВ:

студент 2 курсу ФІОТ

групи ІО-41

Смішний Д. М.

ПЕРЕВІРИВ:

ст. вик. Порєв В. М.

Київ – 2016

**Тема:** Використання у проекті C++ модулів на асемблері

**Мета**: Навчитися створювати програми на С++ з використанням модулів на асемблері.

Варіант: 25

Функція -

Розрядність – 352

**longop.asm**.586

.model flat, c

.data

count dd 1h

count1 dd 0

count2 dd 0

countResultVar dd 0

.code

Add\_352\_LONGOP proc Result:DWORD, B:DWORD, A: DWORD

mov esi, A ;A

mov ebx, B ;B

mov edi, Result ;результат

mov ecx, 352 ; ECX = потрібна кількість повторень

shr ecx, 5

mov edx, 0

clc ; обнулює біт CF регістру EFLAGS

cycle:

mov eax, dword ptr[esi+4\*edx]

adc eax, dword ptr[ebx+4\*edx] ; додавання групи з 32 бітів

mov dword ptr[edi+4\*edx], eax

inc edx

dec ecx ; лічильник зменшуємо на 1

jnz cycle

ret

Add\_352\_LONGOP endp

Sub\_352\_LONGOP proc Result:DWORD, B:DWORD, A:DWORD

mov esi, A ;A

mov ebx, B ;B

mov edi, Result ;результат

mov ecx, 0

mov ecx, 352 ; ECX = потрібна кількість повторень

mov edx, 0

clc ; обнулює біт CF регістру EFLAGS

cycle:

mov eax, dword ptr[esi+4\*edx]

sbb eax, dword ptr[ebx+4\*edx] ; віднімання групи з 32 бітів

mov dword ptr[edi+4\*edx], eax

inc edx

dec ecx ; лічильник зменшуємо на 1

jnz cycle

ret

Sub\_352\_LONGOP endp

Mul\_N32\_LONGOP proc Result: DWORD, B: DWORD, A: DWORD

;push ebp

;mov ebp, esp

mov esi, A ; A

mov ebx, B ; B

mov edi, Result ; результат

mov ecx, 0

@cycle:

mov eax, dword ptr[esi+ 4\*ecx]

mul ebx

add dword ptr[edi+4\*ecx], eax

add dword ptr[edi+4\*ecx+4], edx

inc ecx

cmp ecx, 1

jb @cycle

;pop ebp

ret

Mul\_N32\_LONGOP endp

Mul\_N\_x\_N\_LONGOP proc Result:DWORD, A1:DWORD

;push ebp

;mov ebp, esp

mov esi, A1 ;A

;mov edi, A2 ;A

mov ebx, Result ;результат

mov ecx, 11

@cycle:

push ecx

mov ecx, 11

@cycleIn:

push ecx

mov ecx, count1

mov eax, dword ptr[esi +4 \* ecx]

mov ecx, count2

mul dword ptr[esi + 4 \* ecx]

mov ecx, countResultVar

clc

adc eax, dword ptr[ebx + 4 \* ecx]

mov dword ptr[ebx + 4 \* ecx], eax

adc edx, dword ptr[ebx + 4 \* ecx + 4]

mov dword ptr[ebx + 4 \* ecx + 4], edx

inc count1

inc countResultVar

pop ecx

dec ecx

jnz @cycleIn

inc count2

xor eax, eax

mov count1, eax

mov eax, count2

mov countResultVar, eax

pop ecx

dec ecx

jnz @cycle

xor eax, eax

mov count1, eax

mov count2, eax

mov countResultVar, eax

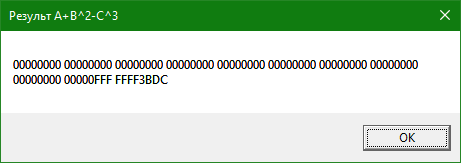
;pop ebp

ret

Mul\_N\_x\_N\_LONGOP endp

end

**Результати роботи**



**Висновок**

В ході лабораторної роботи було розроблено навик створення програми на С++ з використанням модулів на асемблері. Було виконано підрахунок чисел за формулою .