МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин

Звіт

Лабораторна робота №1

З дисципліни: “ Системне програмування ”
Тема: " Особливості програмування з використанням 32-розрядного. "

Варіант 12

Виконав: ст. гр. КІ-38

Папіш Олександр Ростиславович

Керівник: асистент каф. ЕОМ

Козак Н.Б.

ЛЬВІВ 2020

Мета роботи

Ознайомитися з програмною моделлю 32-розрядних мікропроцесорів Intel та оволодіти навиками створення програм, використовуючи 32-розрядний Асемблер

ЗАВДАННЯ

1. Створити, використовуючи мову асемблера мікропроцесорів сімейства x86 Intel, *.exe програму, яка реалізовує обчислення, заданого варіантом виразу. **A = {a[i]}** – наперед заданий масив з **N** чисел цілих чисел. ***c***, ***d*** – цілі константи. **K**, **L** – цілі додатні числа.
2. Переконатися у правильності роботи програми використовуючи VKDebug.
3. Скласти звіт про виконану роботу з приведенням тексту програми.
4. Дати відповідь на контрольні запитання.

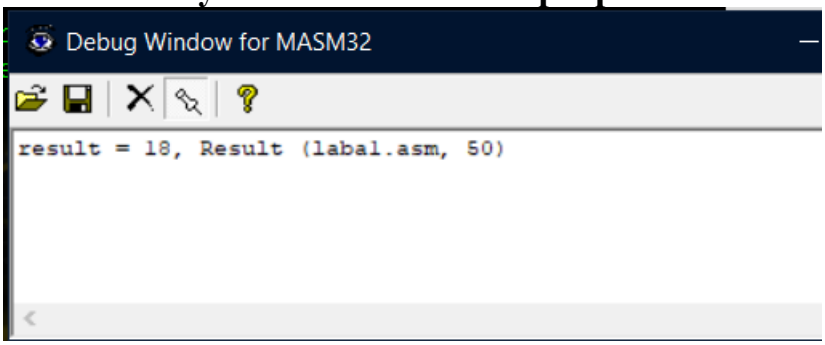
12.	Знайти суму перших **K** додатних елементів **A** за умови **a[i]** >= **c+d**

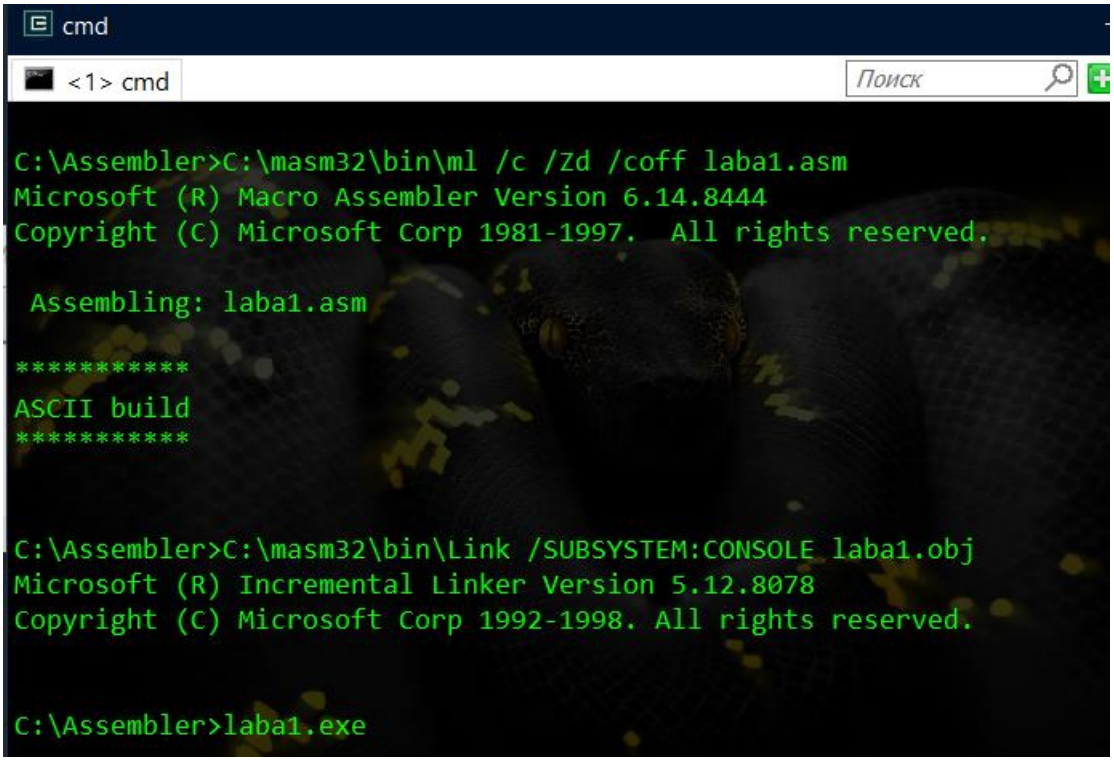
Код програми

```
.586
.model flat, stdcall
option casemap :none

include C:\masm32\include\user32.inc
include C:\masm32\include\windows.inc
include C:\masm32\include\kernel32.inc
include C:\masm32\include\masm32.inc
include C:\masm32\include\debug.inc

includelib C:\masm32\lib\kernel32.lib
includelib C:\masm32\lib\masm32.lib
includelib C:\masm32\lib\debug.lib

.data

 A dd 1,2,3,4,5,1,2,3,4,5 ; масив з додатніми елементами

 K dd 10    ; кількість елементів, які потрібно взяти з масиву
 const_c dd -4    ; числова константа
 const_d dd 8    ; числова константа

k_error db 13,10,'Error! K equal or less then zero!',13,10
```

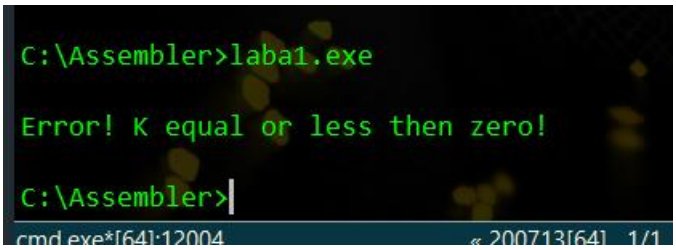
```
result dd 0

.code
start:
    cmp K, 0
    jg kValid
    invoke StdOut, addr k_error
    invoke ExitProcess, 0

    kValid: ; Якщо K більше нуля
        mov edx, const_c
        add edx, const_d

        mov ecx, K
        make_result:
            mov eax, [A+ecx*4-4]
            cmp eax, edx
            jl Ai_l_cd
            cmp eax, 0
            jz Ai_l_cd
            add eax, result
            mov result, eax
            Ai_l_cd:
            dec ecx
            cmp ecx, 0
            jnz make_result

            PrintDec result, "Result"
            invoke ExitProcess, 0

end start
```

Результат виконання програми





Коли К – від’ємне



КОНТРОЛЬНІ ПИТАННЯ

1. Які програмно доступні регістри архітектури IA-32 ви знаєте?
2. У чому полягає різниця між програмними моделями архітектур x86 і IA-32?
3. Як написати програму використовуючи MASM 32?
4. Як відлагодити програму використовуючи MASM 32?
5. Принципи роботи з масивами даних в Асемблері.

Відповіді:

1) eax, ax, ah, al, ebx, bx, bh, bl, ecx, cx, ch, cl, edx, dx, dh, dl, esi, si, edi, di, esp, sp, ebp, bp, cs, ss, ds, es, gs, fs, eflags, flags, eip, ip.

2) Різниця між x86 та IA-32 є суттєвою. Перша архітектура була представленна в процесорі Inter 8086. Шина даних в цьому процесорі була 16 бітна. IA-32 почали використовувати з процесора i368. Шина даних 32 бітна.

3) Для написання програми використовуючи MASM потрібно завантажити макроасемблер Masm32, запустити qeditor.exe. Написати код програми, зберегти файл в розширенні .asm, побудувати exe файл та запустити.

4) Для відлагодження програми використовується бібліотека debug.lib та debug.inc, яка містить в собі макроси відлагодження.

5) Принципи в роботі з масивами даних в Асемблері полягають в адресації памяті. При цьому застосовується адресація за базою з маштабуванням: початкова адреса + база * маштабуючий коефіцієнт.

Висновок: на даній лабораторній роботі я ознайомитися з програмною моделлю 32-розрядних мікропроцесорів Intel та оволодів навиками створення програм, використовуючи 32-розрядний Асемблер.