Міністерство освіти та науки України

Національний університет “Одеська політехніка”

Інститут комп’ютерних систем

Кафедра комп’ютерних інтелектуальних систем та мереж

Лабораторна робота №5

З дисципліни «Машинно-орієнтоване програмування»

16 варіант

Виконав студент

групи АМ221

Моспін Д. О.

Перевірив викладач

Головачова О. В.

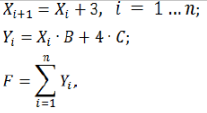
Одеса 2024

ТЕМА: Застосування циклів в мові програмування Assembler.

МЕТА: Вивчити принципи організації циклів в мові програмування Assеmbler.

Завдання: Написати програму обчислення за формулою. Користувач вводить початкове

значення X1, коефіцієнти B, C та індекс n.



Вивести результат:

X1 = , X2 = , … , Xn =

Y1 = , Y2 = , … , Yn =

F = .

Словесний опис:

Код програми:

#include <iostream>

#include <windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int X1, B, C, n;

cout << "Формули за якими виконується програма:" << endl;

cout << "i = 1 ... n;" << endl;

cout << "Х[i+1] = X[i] + 3" << endl;

cout << "Y[i] = X[i] \* B + 4 \* C" << endl;

cout << "F = Y[1] + Y[2] ... + Y[n]" << endl << endl;

// Введення даних

cout << "Введіть початкове значення X1: ";

cin >> X1;

cout << "Введіть коефіцієнт B: ";

cin >> B;

cout << "Введіть коефіцієнт C: ";

cin >> C;

do {

cout << "Введіть індекс n (n > 0): ";

cin >> n;

} while (n <= 0);

if (n > 50) { // Обмеження для запобігання переповненню масиву

cout << "Індекс n повинен бути <= 50!" << endl;

return 1;

}

int X[50] = { 0 }; // Масив для значень X

int Y[50] = { 0 }; // Масив для значень Y

int F = 0; // Підсумкова сума

\_\_asm {

mov eax, X1 // Завантажуємо початкове значення X1

mov ebx, B // Завантажуємо коефіцієнт B

mov edx, C // Завантажуємо коефіцієнт C

lea esi, X // Адреса масиву X в регістрі esi

lea edi, Y // Адреса масиву Y в регістрі edi

mov ecx, n // Кількість ітерацій у регістрі ecx

mov[F], 0 // Ініціалізація суми F

loop\_start:

//Збереження поточного значення X у масив X

mov[esi], eax // Зберігаємо значення X у масиві X

add esi, 4 // Переходимо до наступного елемента масиву X (int займає 4 байти)

// Y[i] = X[i] \* B + 4 \* C

mov ebx, B // Завантажуємо коефіцієнт B

imul ebx, eax // ebx = X[i] \* B

mov edx, C // Завантажуємо коефіцієнт C

imul edx, 4 // edx = 4 \* C

add ebx, edx // ebx = X[i] \* B + 4 \* C

mov[edi], ebx // Зберігаємо значення Y[i] у масиві Y

add edi, 4 // Переходимо до наступного елемента масиву Y

// Додаємо Y[i] до підсумкового значення F

add[F], ebx // F = F + Y[i]

//Обчислюємо X[i + 1] = X[i] + 3

add eax, 3 // eax = X[i] + 3

loop loop\_start // Повторюємо цикл

}

// Виведення результатів

cout << "Результати:\n";

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

cout << "X" << (i + 1) << " = " << X[i] << ", ";

cout << "Y" << (i + 1) << " = " << Y[i] << endl;

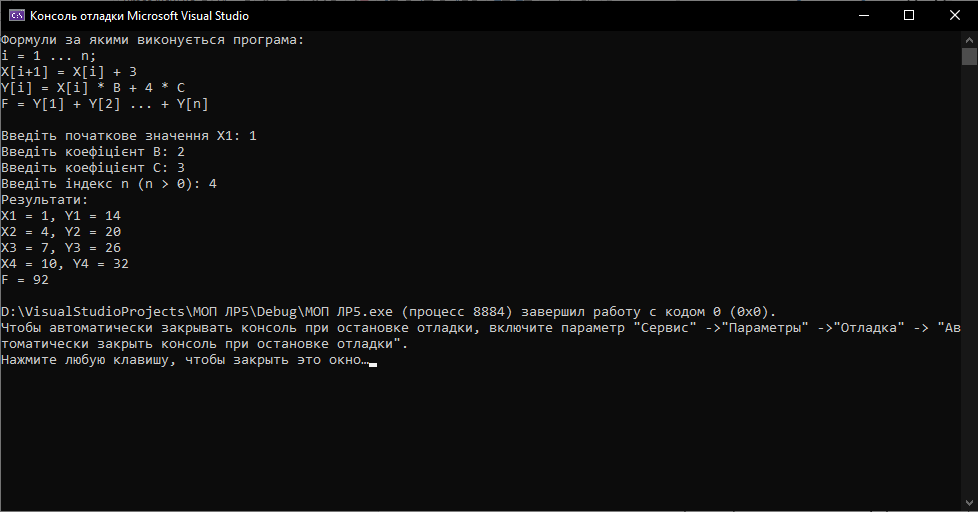
}

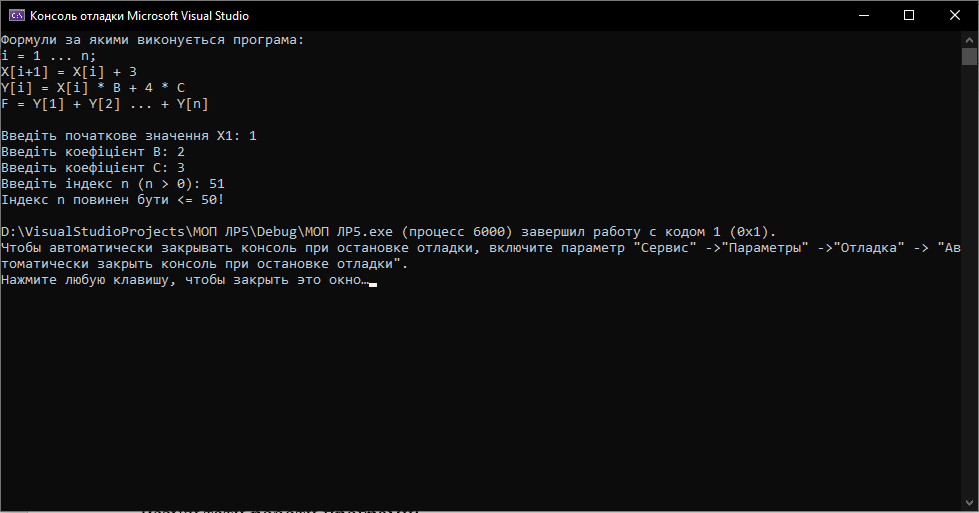
cout << "F = " << F << endl;

return 0;

}

Результати роботи програми:





Висновок:

У даній лабораторній роботі, я вивчив принципи організації циклів в мові програмування Assembler.