Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Инженерно-психологическое

обеспечение информационных технологий»

Учебная дисциплина «Основы алгоритмизации и программирования»

Отчет

по лабораторной работе No1

«Линейные алгоритмы»

Вариант 25

Подготовил:

Подрябинкин З.Д. 410902

Проверил: Усенко Ф.В.

Минск 2024

Цель: Сформировать умения разрабатывать программы с использованием линейных алгоритмов.

Задание: (Вариант **25**)Составить программу для расчета по двум формулам. Предусмотреть ввод значения угла в градусной мере. Организовать красивый вывод результата работы программы. Правильность работы программы легко проверить - результаты вычисления по обеим формулам должны совпадать.

Z1= Z2=

Далее представлены код, схема и результат работы

Программы для задания

(Замечание: для выполнения условия из задания понадобилось использовать условный оператор. А раз придётся выйти за рамки темы, то для удобства проверки присутствую цикл)

#include <iostream>

#define PI -2/3

#include <cstdlib>

#include <cmath>

using namespace std;

// Число Пи для вычислений

int main()

{

int count = 0;

cout << "The program for calculating formula was launched. \n \n";

while (count != 1)

{

float NumbIn; // Основная переменная

float Res1;

float Res2;

float FRes; // Результат

float GetNomber;

cout << "### \n Enter your number in degrees: ";

cin >> GetNomber; // Получение числа от пользователя

NumbIn = GetNomber;

cout << "### \n \n";

float Temp = NumbIn - ( (int)NumbIn / 360 ) \* 360; // Тангенс симметричен относительно начала координат и цикличен, для удобства приведём градусную меру к значению между -360 и +360

NumbIn = NumbIn \* (PI / 180); // Перевод градусов в радианы для мат функций c++

Res1 = ((2 \* tan(NumbIn / 2)) / (1 + pow(tan(NumbIn / 2), 2))); //расчёт по первой формуле

if (abs(Temp) <= 90 or abs(Temp) >= 270) // Расчёт по второй формуле, с учётом знака

{

Res2 = ((tan(NumbIn)) / (sqrt(1 + pow(tan(NumbIn), 2))));

}

else

{

Res2 = ((tan(NumbIn)) / (-sqrt(1 + pow(tan(NumbIn), 2))));

}

FRes = round(((Res1 + Res2) / 2) \* 100000) / 100000; //Нахождение среднеарифметического и его округление до статысячных

cout << " Result caculated with first formula: " << Res1 << "\n"; // Вывод уточняющей информации

cout << " Result caculated with first formula: " << Res2 << "\n";

cout << " Awerage rounded result: " << FRes << "\n \n";

cout << "### \n Final result: " << FRes << "\n### \n \n"; // Результат для пользователя

cout << " If you want to exit the program, enter 1. \n If you want to continue, enter any other number. \n"; // завершение программы или повторный цикл для удобства работы

cout << " Answer: ";

cin >> count;

}

cout << "Programm stoped. \n";

system("pause");

return 0;

}

Код программы

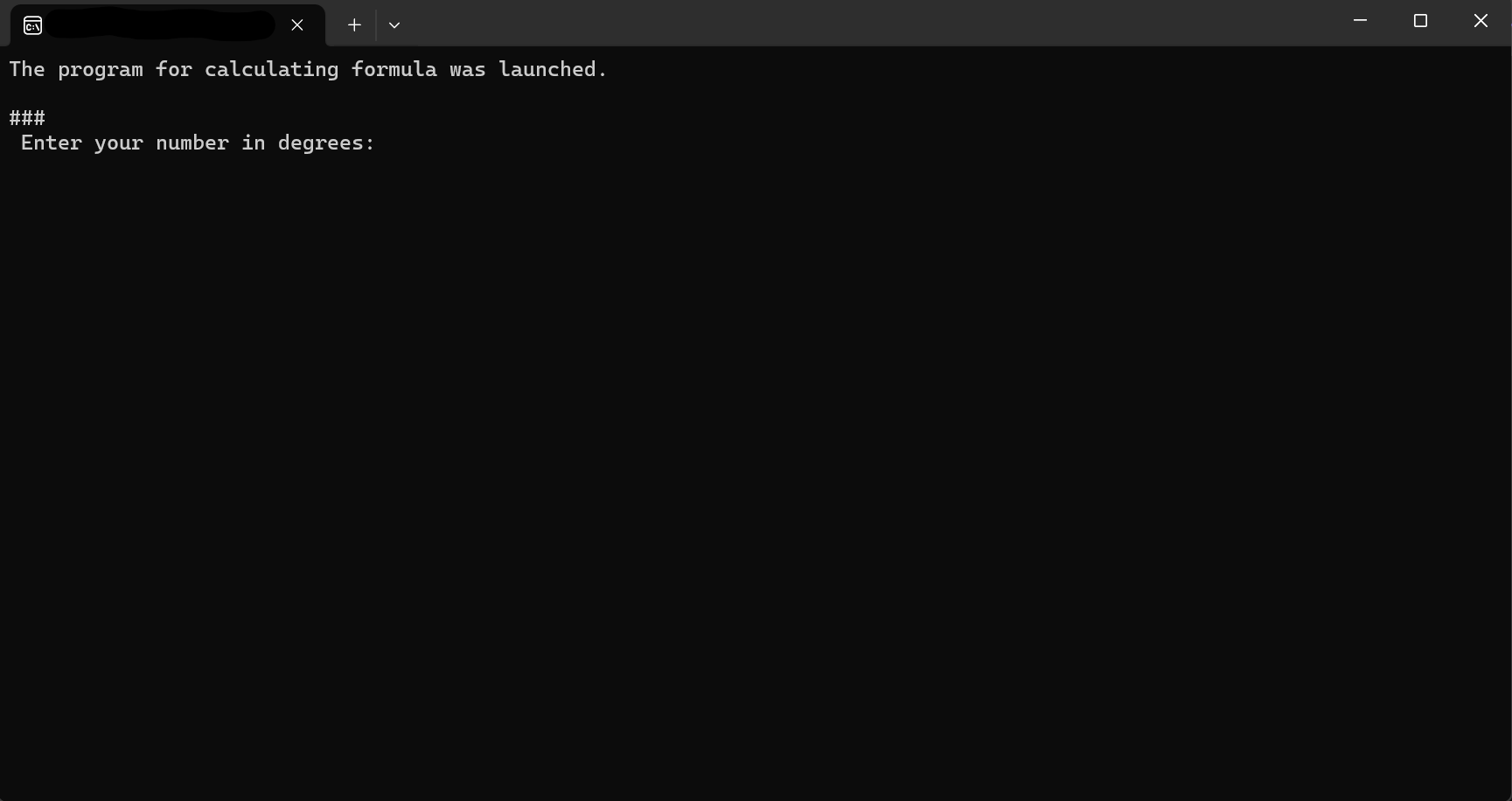


Рисунок 1 – Скриншот запроса данных у пользователя программой

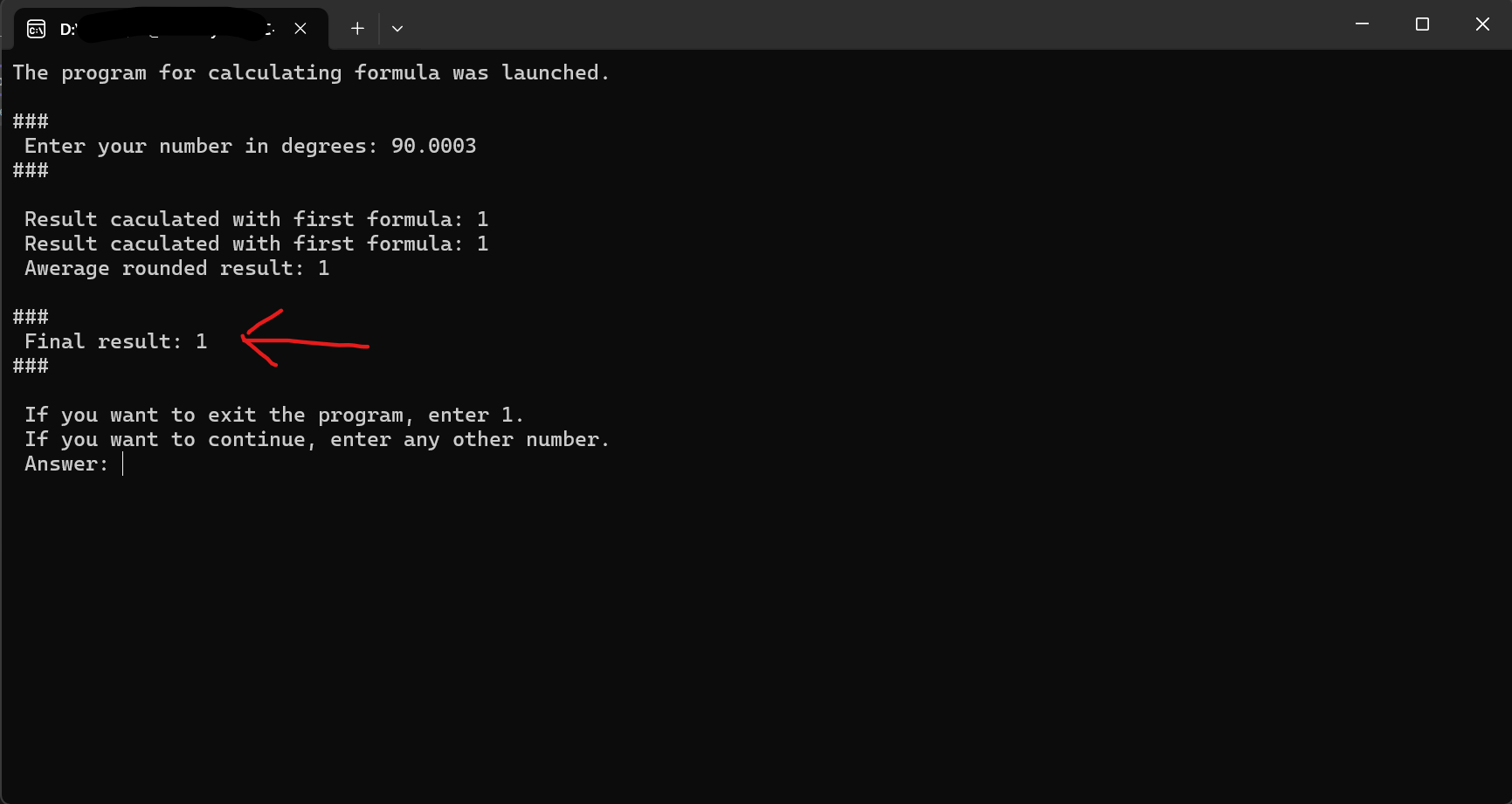


Рисунок 2 – Скриншот результата при вводе 90,0003⁰

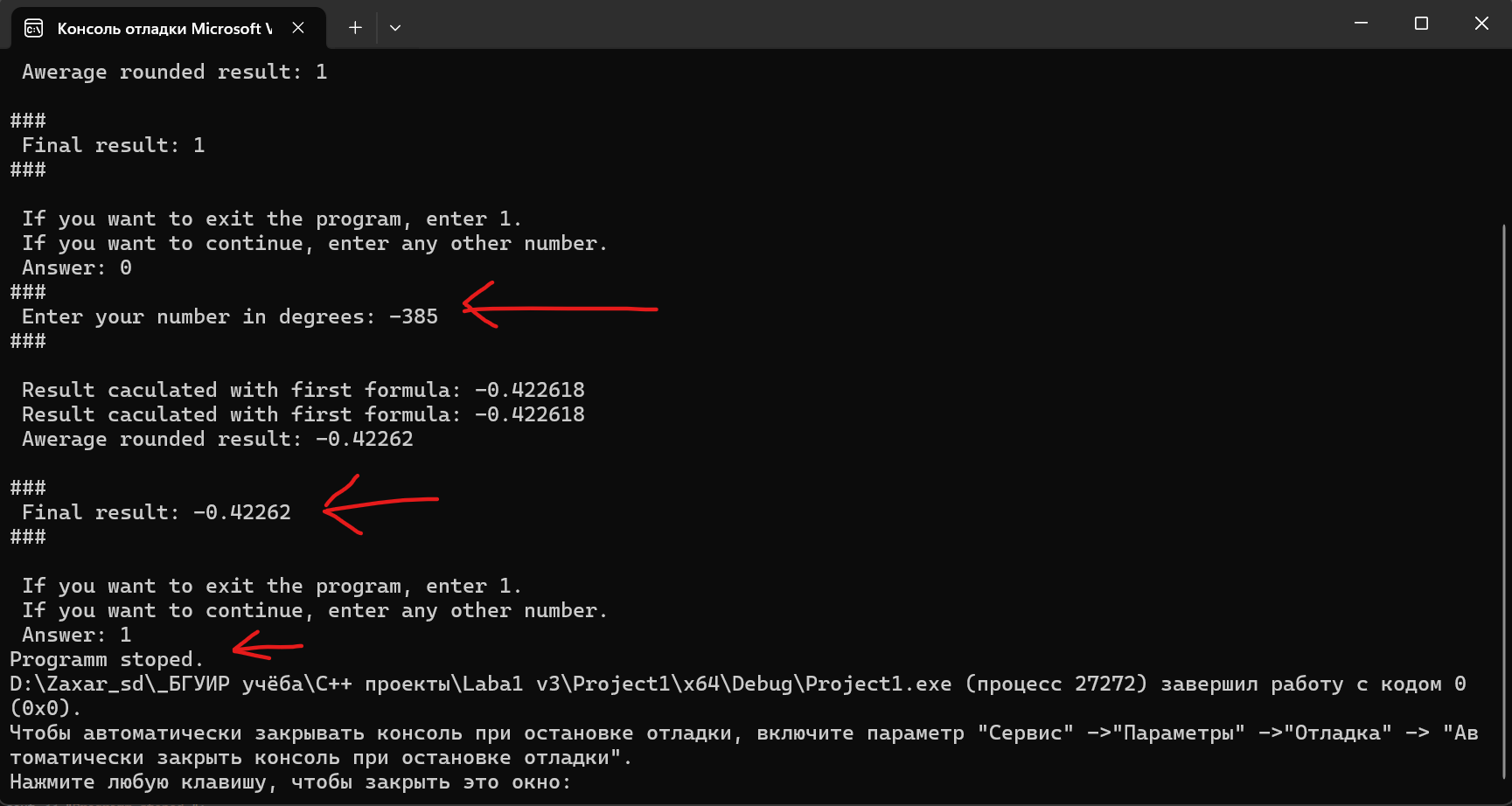


Рисунок 3 – Скриншот с демонстрацией ввода отрицательного большого числа и выхода из программы

Вывод: в ходе выполнения работы создана программа производящее вычисление по заданным формулам. Итоговый результат вычисляется как средне арифметическое результатов вычисления по двум заданным формулам, округлённое до 5 знака после запятой. В ходе выполнения работы появилась необходимость ввести условный оператор поскольку одна из формул даёт результат с неопределённым знаком (Z2). Помимо этого, для удобства использования и отладки был применён оператор цикла.