**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего образования**

**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет №3

«Системы управления, информатика и электроэнергетика»

Кафедра 304

**Отчет по лабораторной работе №2**

по учебной дисциплине «Информатика»

на тему «Табулирование функций»

Вариант № 15

Руководитель работы

\_\_\_\_\_\_\_\_\_\_\_\_ Давыдкина Е. А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2021 г.

\_\_\_\_\_\_\_\_\_\_\_\_ Секретарев В. Е.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2021 г.

Исполнитель

обучающийся группы М30-110Б-21

\_\_\_\_\_\_\_\_\_\_\_\_ Осминнов Н. М.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

\_\_\_\_\_\_\_\_\_\_\_\_Пысларь А. И.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

**Содержание.**

1. Задание……………………………………………….......................................................................3
2. Блок-схема………………………………………………………………………………………………………………4
3. Псевдокод……………………………………………………………………………………………………………….5
4. Код программы……………………………………………………………………………………………………….7
5. Тесты программы…………………………………………………………………………………………………...10
6. Вывод…………………………………………………………………………………………………………………….15

**Задание.**

Кафедра 304 Курс: ИНФОРМАТИКА

Задание 2: ***Табулирование функций.***

ВАРИАНТ № 15

Разработать, отладить и протестировать программу, которая разбивает отрезок [A;B] на N интервалов; в зависимости от значений A, B и N вычисляет величину шага h = (B - A) / N, с которым происходит приращение аргумента xi = A + i \* h, и в каждой точке xi печатает значения функций:

F(xi) = (x - 2) 3

(x + 1) 3

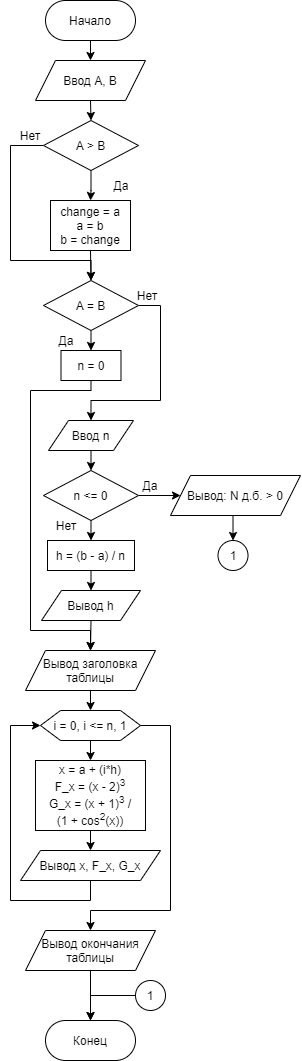
G(xi) =

1 + cos 2 (x)

Результаты вычислений представить в виде таблицы.

В качестве одного из вариантов исходных данных взять: A = 0, B =  N = 10.

**Блок-схема.**

****

**Псевдокод.**

**НАЧАЛО**

**ВЕЩ**

a // начало отрезка

b // конец отрезка

h //шаг между точками

change // переменная для замены a и b местами

x // Полученное значение x

F\_x // значение функции F(x) в точке x

G\_x// значение функции G(x) в точке x

**ЦЕЛ**

n // количество отрезков

**ВВОД** a

**ВВОД** b

//Входной контроль

**ЕСЛИ** a > b

**ТО**

change = a

a = b

b = change

**КОНЕЦ**

**ЕСЛИ** a = b

**ТО**

n = 0

**ИНАЧЕ**

**ВВОД**  n

**ЕСЛИ** n <= 0

**ТО**

**ЗАВЕРШИТЬ РАБОТУ АЛГОРИТМА**

**КОНЕЦ**

h = ( b - a ) / n

**КОНЕЦ**

//Вывод заголовка таблицы

**ВЫВОД** «╔══╦══╦══╗»

**ВЫВОД** «║ “x” ║ “F(x)” ║ “G(x)” ║»

//Расчет и вывод данных в таблицу

**ВЕЩ** I = 0

**ДЕЛАТЬ ПОКА** I <= n

x = a + (i\*h)

F\_x = (x-2)3

G\_x = ((x + 1)3) / (1 + (cos(x))2)

//Печать таблицы с данными

// ПЕРЕХОД НА СЛЕДУЮЩУЮ СТРОКУ ТАБЛИЦЫ

**ВЫВОД:** «╠══╬══╬══╣»

// ВЫВОД СТРОКИ ТАБЛИЦЫ

**ВЫВОД:** «║ x ║ F\_x ║ G\_x ║»

i = I + 1

**КЦ**

//Вывод окончания таблицы

**ВЫВОД:** «╚══╩══╩══╝»

**КОНЕЦ**

**Код Программы.**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Курс ИНФОРМАТИКА \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Project type :Win64 Console Application \*

\*Project name :Lab1 \*

\*File name :Лабораторная работа.exe \*

\*Language :CPP, MSVS 2021 \*

\*Programmers :Осминов Никита Михайлович, Пысларь Александр Игоревич, М3О-110Б-21 \*

\*Modified By : \*

\*Created :17.09.2021 \*

\*Last revision:01.12.2021 \*

\*Comment :Табулирование функций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <iomanip>

int main()

{

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    ------ИНИЦИАЛИЗЫЦИЯ ПЕРЕМЕННЫХ------

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    using namespace std;

    float a;                                  //Начало отрезка

    float b;                                  //Конец отрезка

    float h = 0;                              //Шаг между точками

    float change;                             //Переменная для смены местами А и В

    float x;                                  //Полученное значение x

    float F\_x, G\_x;                           //Функции в точке x

    int n;                                    //Количество отрезков

    setlocale(LC\_ALL, "Russian");

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    ----ПОДГОТОВКА ЭЛЕМЕНТОВ ПСЕВДОГРАФИКИ----

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    const char VerticalLine = (char)186;      //"║" - Вертикальная линия

    const char HorizontalLine = (char)205;    //"═" - Горизонтальная линия

    const char TopLeftCorner = (char)201;     //"╔" - Левый верхний угол

    const char TopRightCorner = (char)187;    //"╗" - Правый верхний угол

    const char BottomLeftCorner = (char)200;  //"╚" - Левый нижний угол

    const char BottomRightCorner = (char)188; //"╝" - Правый нижний угол

    const char LeftJunction = (char)185;      //"╣" - вертикальная "труба влево"

    const char RightJunction = (char)204;     //"╠" - вертикальная "труба вправо"

    const char BottomJunction = (char)203;    //"╦" - Горизонтальная вниз "труба"

    const char TopJunction = (char)202;       //"╩" - Горизонтальная вниз "труба"

    const char CrossJunction = (char)206;     //"╬" - пересечение линий

    const char Space = (char)32;

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    -----ВВОД И ПРОВЕРКА ДАННЫХ-----

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    cout << "Введите A: ";

    cin >> a;

    cout << "Введите B: ";

    cin >> b;

    if (a > b) //Если A > B меняем значения переменных А и В местами.

    {

      cout << "A>B => меняем их значения местами." << endl;

      change = a;

      a = b;

      b = change;

    }//конец if

    if (a == b)  //Если а = b то

    {

       n = 0;    //Отрезков нет

       cout << " A = B => N = 0" << endl;

    }

    else {

        cout << "Введите N: ";

        cin >> n;

        if (n <= 0)         //Проверка значения N

        {

            cout << "N д.б. > 0" << endl;

            exit(1);

        }//конец if

        h = (b - a) / n;    //Считаем шаг

        cout << "H = " << h << endl;

    }//конец if

    setlocale(LC\_ALL, "C"); //Отключаем русский язык во избежание курьезов.

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    ПЕЧАТЬ ВЕРХУШКИ ТАБЛИЦЫ И ОБОЗНАЧЕНИЙ ДАННЫХ

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    //╔══╦══╦══╗

    cout << TopLeftCorner << setfill(HorizontalLine) << setw(20)

         << BottomJunction

        << setw(20) << BottomJunction

        << setw(20) << TopRightCorner << endl;

    //║ x ║ F(x) ║ G(x) ║

    cout << VerticalLine << setfill(Space)

        << setw(19) << right << "x" << VerticalLine

        << setw(19) << right << "F(x)" << VerticalLine

        << setw(19) << right << "G(x)" << VerticalLine << endl;

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    ---ОСНОВНОЙ АЛГОРИТМ---

    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    for (int i = 0; i <= n ; i++ )

    {

        x = a + (i\*h);                       //Считаем xi значения функций в точке x.

        F\_x = pow((x - 2), 3);

        G\_x = pow((x + 1), 3) / (1 + pow((cos(x)), 2));

        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

        -----ПЕЧАТЬ ТАБЛИЦЫ С ДАННЫМИ------

        \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

        // ╠══╬══╬══╣

        cout << RightJunction << setfill(HorizontalLine)

            << setw(20) << CrossJunction << setw(20)

                        << CrossJunction << setw(20)

            << LeftJunction << endl;

        // ║ x ║ F\_x ║ G\_x ║

        cout << VerticalLine << setfill(Space) << setw(19)

            << right << x << VerticalLine << setw(19)

            << right << F\_x << VerticalLine << setw(19)

            << right << G\_x << VerticalLine << endl;

    }//конец for

    // ╚══╩══╩══╝

    cout << BottomLeftCorner << setfill(HorizontalLine) << setw(20)

        << TopJunction << setw(20)

        << TopJunction << setw(20)

        << BottomRightCorner << endl;

}

**Тесты программы.**

**1.** Некорректные тесты:

**1.1** Цель: проверить работоспособность программы из некорректной области.

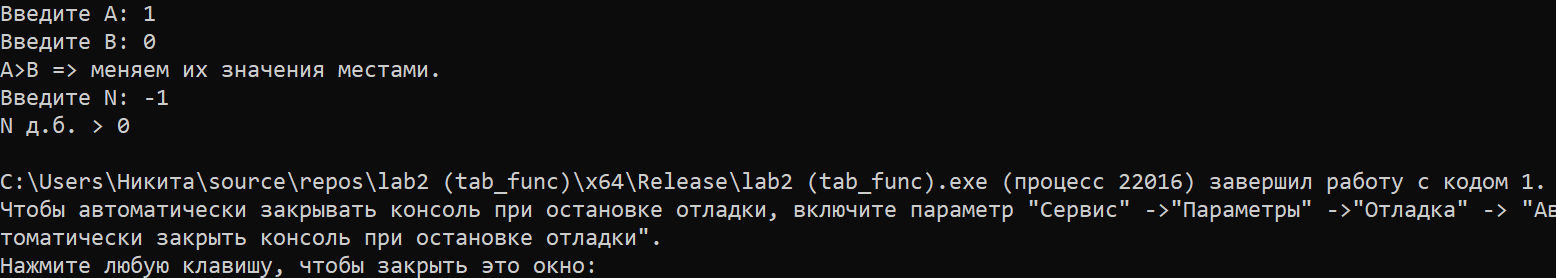
Исходные данные: A = 1; B = 0; N = -1

Ожидаемый результат:

A>B => меняем их значения местами

N д.б. > 0

Закрытие программы

Полученный результат: 

Вывод по тесту: полученный результат совпал с ожидаемым.

**1.2** Цель: проверить работоспособность программы из некорректной области.

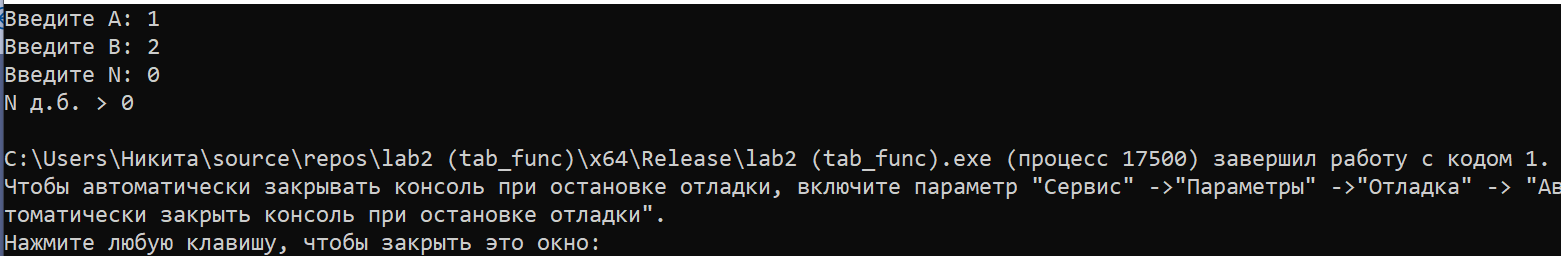
Исходные данные: A = 1; B = 2; N = 0

Ожидаемый результат:

N д.б. > 0

Конец работы программы.

Полученный результат:



Вывод по тесту: полученный результат совпал с ожидаемым.

**2.** Корректные тесты:

**2.1** Цель: проверить работоспособность программы, попадая в основной цикл.

Исходные данные: A = 1; B = 2; N = 3

Ожидаемый результат:

H = 0.(3) = 1/3;

X1 = 1;

F(x1) = (1 - 2)3 = -1;

G(x1) = ((1+1)3)/(1 + cos2(1)) = 6.1923

X2 = 1+0.(3) =1.(3) = 4/3;

F(x2) = (1.(3) – 2)3 = -0.296296 G(x2)=((1.(3)+1)3)/(1+cos2(1.(3))) = 12.0376

X3 = 1.(3) + 0.(3) = 1.(6) = 5/3

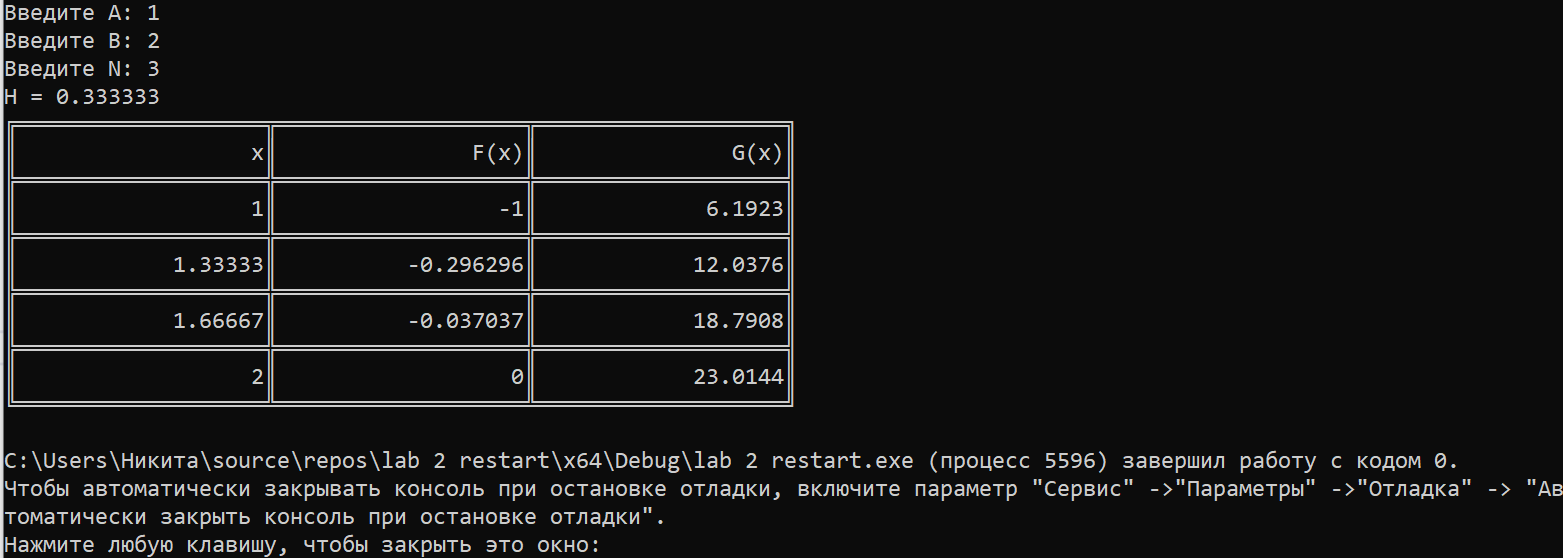
F(x3) = (1.(6)-2)3=-0.37037

G(x3) = ((1.(6)+1)3)/(1+cos2(1.(6))) = 18.7908

X4 = 2 = 6/3

F(x4) = (2 - 2)3 = 0 G(x4) = ((1.(9)+1)3)/(1+ ((cos(1.(9)))2) ~23.0144

Полученный результат:



Вывод: результат совпал с ожидаемым.

**2.2** Цель: проверить работоспособность программы, попадая в основной цикл.

Исходные данные: A = 1; B = 2; N = 2

Ожидаемый результат: H = (2-1)/2 = 0.5

x1 = 1; F(x1) = (1-2)3 = -1

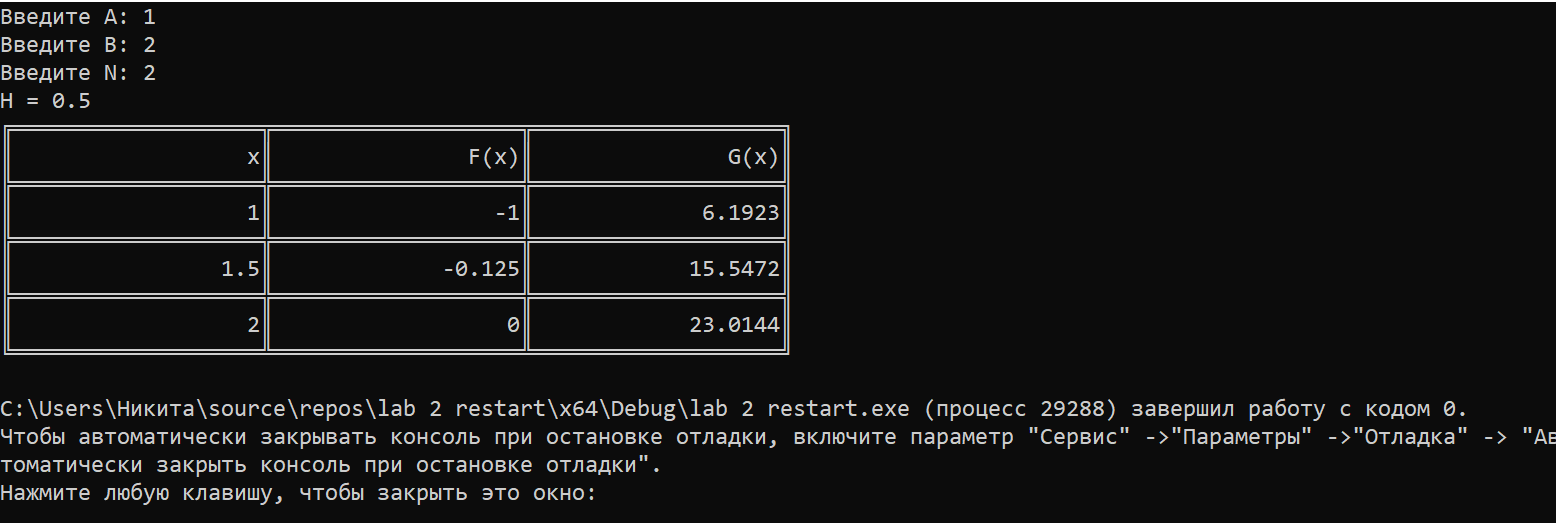
G(x1) = ((1 + 1)3)/(1+(cos2(1)) = 6.1923

X2 = 1+0.5 =1.5 F(x2) = (1.5-2)3=-0.125 G(x2) = ((1.5+1)3)/(1+(cos2(1.5)) = 15.5472

X3 = (1.5 + 0.5) = 2

F(x3) = (2-2)3 = 0 G(x3) = ((2 + 1)3)/(1+(cos2(2)) = 23.0144

Полученный результат:



Вывод: результат совпал с ожидаемым.

**2.3** Цель: проверить работоспособность программы из некорректной области.

Исходные данные: A = 0; B = 0; N = 9

Ожидаемый результат:

A = B => N = 0

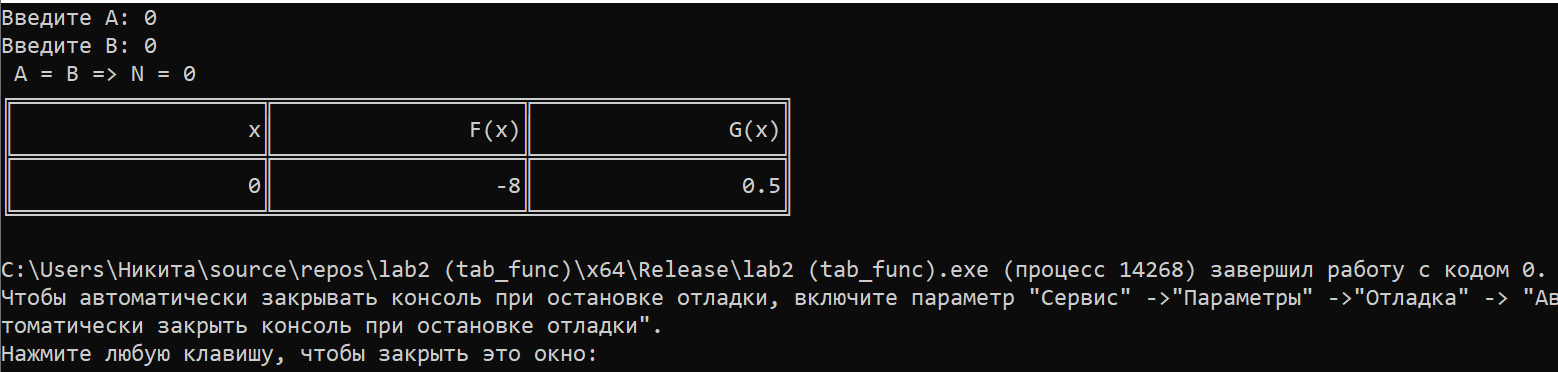
Вывод таблицы с данными из одной точки.

x = 0

F(x) = (0-2)3=-8

G(x)=(0+1)3/(1+ cos2(0)) = 0.5

Полученный результат:



Вывод по тесту: полученный результат совпал с ожидаемым.

**2.4** Цель: проверить работоспособность программы, попадая в основной цикл.

Исходные данные: A = 0; B = π; N = 10

Ожидаемый результат: H = (π -0)/10 = 0.31415

x1 = 0; F(x1) = (0-2)3 = -8

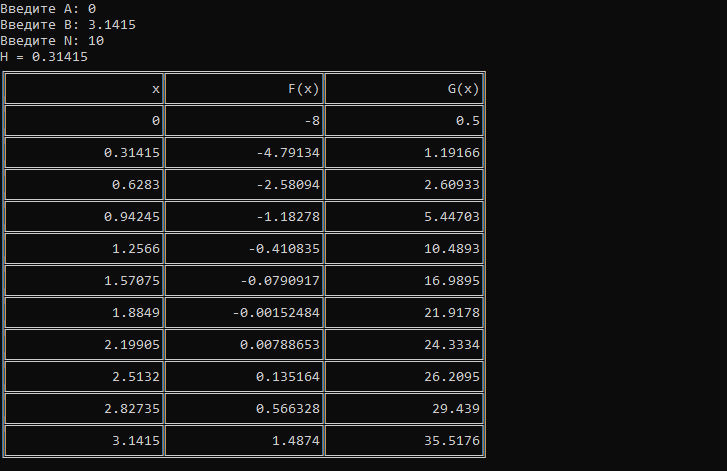
G(x1) = ((0 + 1)3)/(1+(cos2(0)) = 0,5

X2 = 0+0.31415 =0,31415 F(x2) = (0.31415-2)3=-4,79134 G(x2) = ((0.31415 +1)3)/(1+(cos2(0.31415)) = 1.19166

X3 = (0.31415 + 0.31415) = 0,6283

F(x3) = (0,6283 - 2)3 = -2,58094 G(x3) = ((0,6283 + 1)3)/(1+(cos2(0,6283)) = 2,60933

Полученный результат:



Вывод: результат совпал с ожидаемым.

**Вывод по работе.**

Разработка программы завершена на том основании, что:

1. Полученные результаты совпали с ожидаемыми.

2. считаем набор тестов полным.