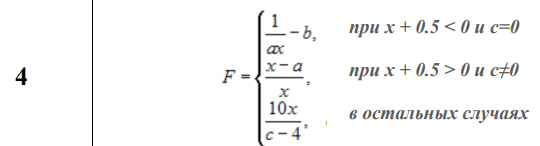
**30333-2 Калевич Сергей Михайлович Лабораторная работа №3 вариант 4**

**ЗАДАНИЕ 1. Использование оператора WHILE.**

Вычислить и вывести на экран в виде таблицы значения заданной функции F от -1 до 1 с шагом 0.05. Значения аргументов функции a, b и c вводятся с клавиатуры.

Предусмотреть сообщение об ошибке в случае исключительной ситуации. Для ввода/вывода используйте библиотеку **stdio**. Результат отобразить в виде таблицы:

--------------------- 

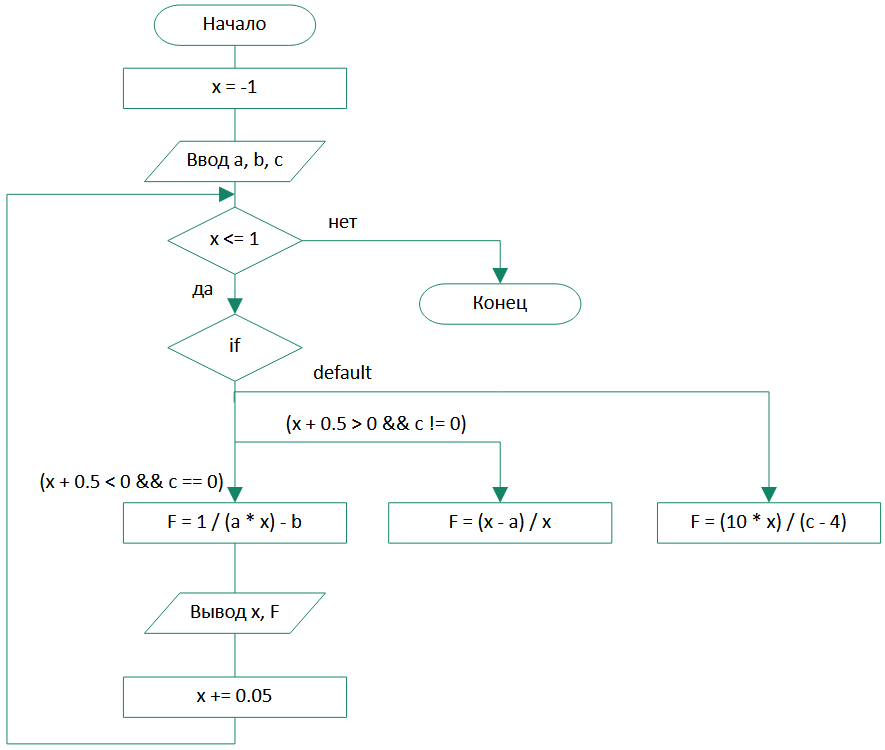
| x | F |

---------------------

| -1.00 | 2.80 |

| -0.95 | 2.60 |

---------------------



**Ход выполнения:**

lab3\_task1\_var4.cpp

// Lab #3

// Task #1

// Variant #4

#include <stdio.h>

#include <Windows.h>

#include <cmath>

using namespace std;

bool isZero = false;

double absEpsilon = 1e-12;

double relEpsilon = 1e-8;

bool approximatelyEqualAbsRel(double a, double b, double absEpsilon, double relEpsilon);

void isZeroDivision(double num);

double fXPlusPointFiveLessZeroAndCEquZero(double a, double b, double x)

{

isZeroDivision(a \* x);

if (isZero) return 0;

return 1 / (a \* x) - b;

}

double fXPlusPointFiveGtrZeroAndCNotZero(double a, double x)

{

isZeroDivision(x);

if (isZero) return 0;

return (x - a) / x;

}

double fDefault(double c, double x)

{

isZeroDivision(c - 4);

if (isZero) return 0;

return (10 \* x) / (c - 4);

}

double fFunc(double a, double b, double c, double x)

{

if (x + 0.5 < 0 && c == 0)

return fXPlusPointFiveLessZeroAndCEquZero(a, b, x);

if (x + 0.5 > 0 && c != 0)

return fXPlusPointFiveGtrZeroAndCNotZero(a, x);

return fDefault(c, x);

}

void printLine(int lineLength)

{

for (int i = 0; i < lineLength; i++)

printf("-");

printf("\n");

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float a{};

float b{};

float c{};

double x{ -1 };

printf("Введите три числа\n");

scanf\_s("%f%f%f", &a, &b, &c);

printLine(25);

printf("| x | F |\n");

printLine(25);

while (x < 1.000001)

{

printf("|%9.2f |%9.2f |\n", x, fFunc(a, b, c, x));

if (isZero)

{

printf("Деление на ноль\n\n");

isZero = false;

}

x += 0.05;

}

printLine(25);

system("pause");

return 0;

}

bool approximatelyEqualAbsRel(double a, double b, double absEpsilon, double relEpsilon)

{

double diff = fabs(a - b);

if (diff <= absEpsilon)

return true;

return diff <= ((fabs(a) < fabs(b) ? fabs(b) : fabs(a)) \* relEpsilon);

}

void isZeroDivision(double num)

{

if (approximatelyEqualAbsRel(num, 0, absEpsilon, relEpsilon))

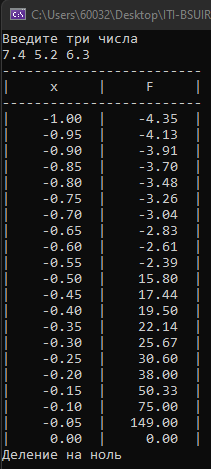
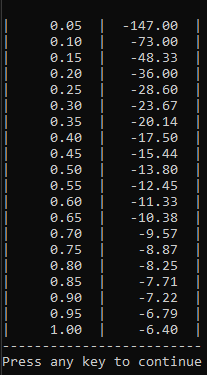
{

isZero = true;

}

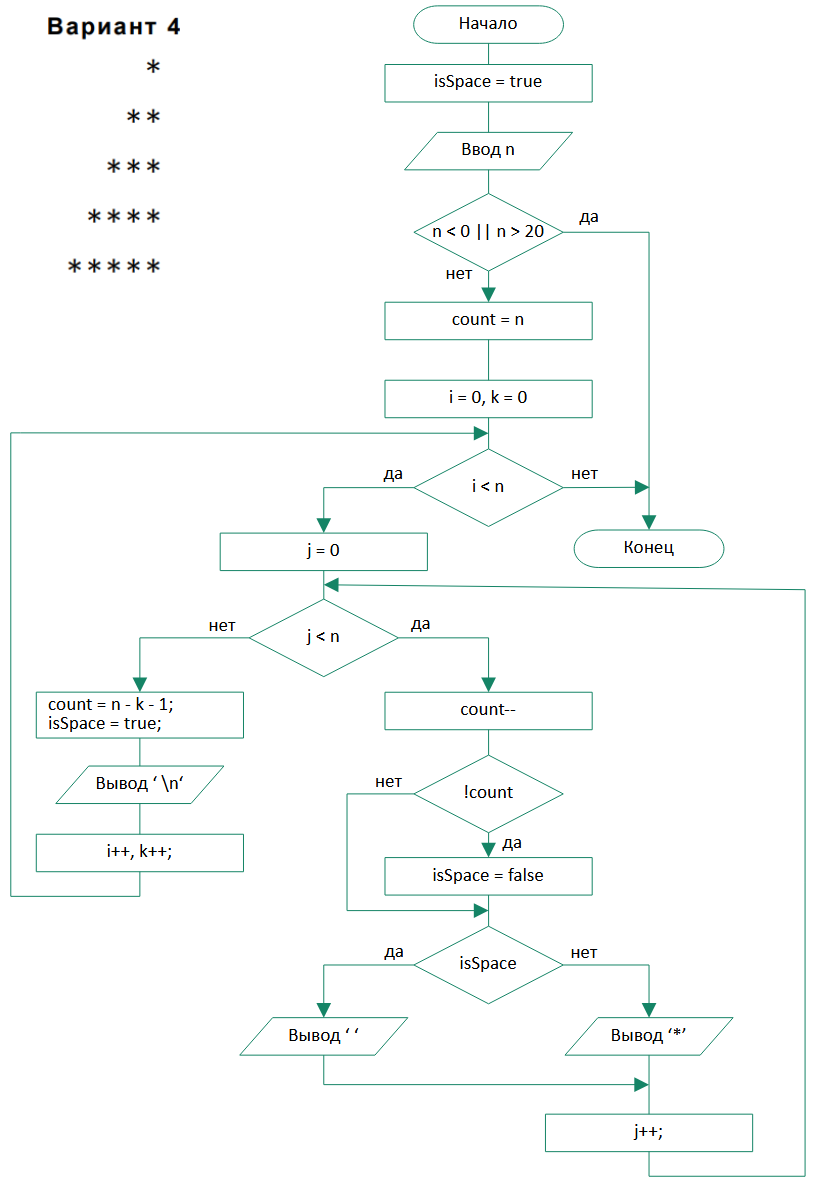
}

**Результат:**

**ЗАДАНИЕ 2. Использование оператора for.**

Пользователь вводит число n (в диапазоне от 0 до 20). Сформировать соответствующую фигуру из звездочек (символов \*). Ваша фигура должна быть вписана в квадрат n\*n символов.



**Ход выполнения:**

lab3\_task2\_var4.cpp

// Lab #3

// task #2

// variant #4

#include <iostream>

#include <Windows.h>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

bool isSpace = true;

int n{};

cout << "Введите число от 0 до 20: ";

cin >> n;

if (n < 0 || n > 20)

{

cout << "Число должно быть в диапозоне от 0 до 20" << endl;

system("pause");

return;

}

int count = n;

for (int i = 0, k = 0; i < n; i++, k++)

{

for (int j = 0; j < n; j++)

{

count--;

if (!count) isSpace = false;

if (isSpace) cout << ' ';

else cout << '\*';

}

count = n - k - 1;

isSpace = true;

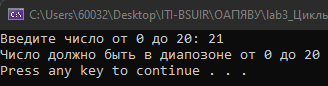
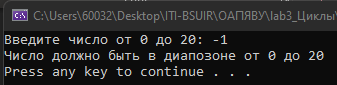
cout << "\n";

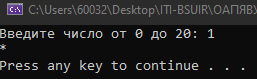
}

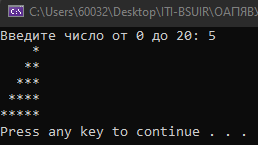
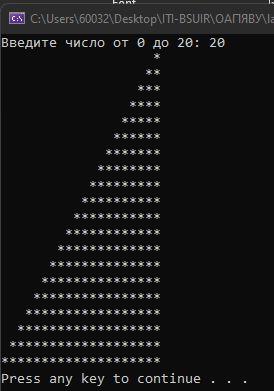
system("pause");

}

**Результат:**

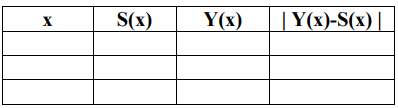


**ЗАДАНИЕ 3.**

Напишите программу, которая выведет на экран таблицу с данными в виде:

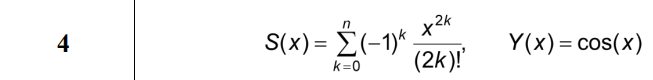


Значение аргумента x изменяется от **a** до **b** с шагом **h**.

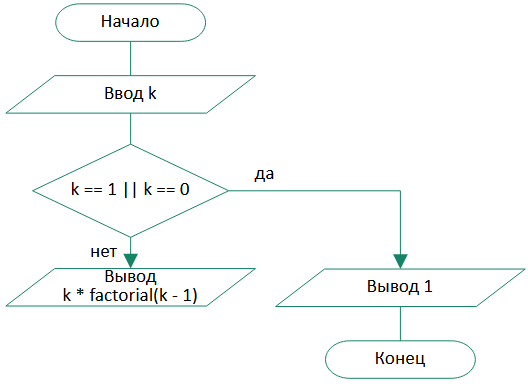
Для каждого **x** рассчитать значения функции **Y(x)**, сумму ряда **S(x)** и модуль разницы **|Y(x)-S(x )|**.

Так как значение **S(x)** является рядом разложения функции **Y(x)**, то значения **S** и **Y** для текущего аргумента **x** должны совпадать в целой части (в двух-четырех позициях после десятичной точки).

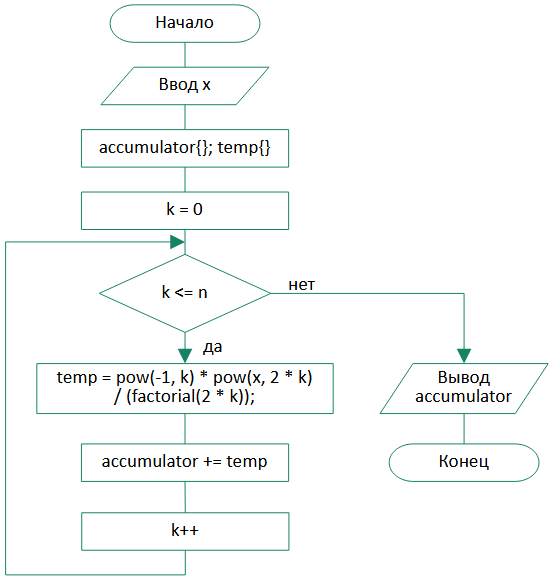
Значения переменных задать статически в коде: **a = 0.1; b = 1.0; h = 0.1; n=20**.



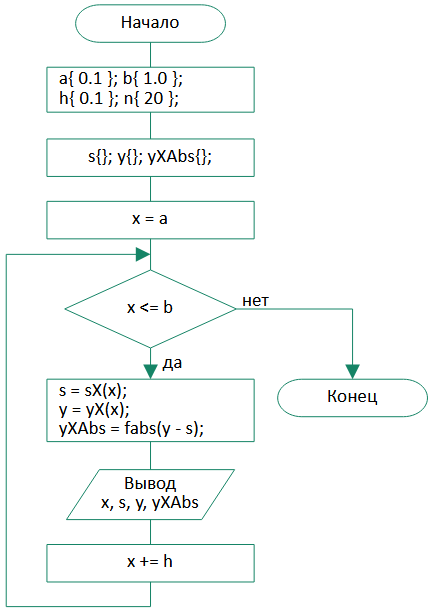
**Факториал:**



**S(x):**



**main():**



**Ход выполнения:**

lab3\_task3\_var4.cpp

// Lab #3

// Task #3

// Variant #4

#include <iostream>

#include <cmath>

using namespace std;

const int lineLength = 85;

const double a{ 0.1 };

const double b{ 1.0 };

const double h{ 0.1 };

const int n{ 20 };

void printLine()

{

printf(" ");

for (int i = 0; i < lineLength; i++)

printf("-");

printf("\n");

}

void printData(double x, double s, double y, double ys)

{

printf(" |%11.1f%9c|%13.4f%7c|%13.4f%7c|%13.4f%7c|\n",

x, ' ', s, ' ', y, ' ', ys, ' ');

printLine();

}

long long factorial(int k)

{

if (k == 1 || k == 0)

{

return 1;

}

else

{

return k \* factorial(k - 1);

}

}

double yX(double x) { return cos(x); }

double sX(double x)

{

double accumulator{};

double temp{};

for (int k = 0; k <= n; k++)

{

temp = pow(-1, k) \* pow(x, 2 \* k) / (factorial(2 \* k));

accumulator += temp;

}

return accumulator;

}

int main()

{

printLine();

printf(" |%10c%10c|%12s%8c|%12s%8c|%16s%4c|\n",

'x', ' ', "S(x)", ' ', "Y(x)", ' ', "| Y(x)-S(x) |", ' ');

printLine();

double s{};

double y{};

double yXAbs{};

for (double x = a; x <= b; x += h)

{

s = sX(x);

y = yX(x);

yXAbs = fabs(y - s);

printData(x, s, y, yXAbs);

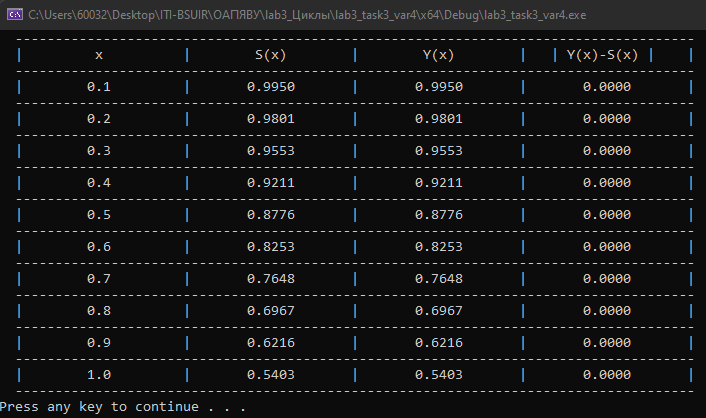
}

system("pause");

return 0;

}

**Результат:**



Выводы

Изучены и применены на практике приемы использования операторов цикла for, while, do – while. На их основе получены навыки программирования циклических алгоритмов.