МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ГРАЖДАНСКОЙ АВИАЦИИ  
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

**Лабораторная работа №1**

**«Основы работы на алгоритмическом языке С++»**

По дисциплине: «Объектно-ориентированное программирование»

Вариант 9

Выполнил:

студент группы ПМ4-1

Фейзуллин К.М.

***Цель работы.***

Целью данной лабораторной работы является освоение синтаксиса языка С++, включая:

- объявление констант и переменных;

- правила записи и вычислений арифметических и логических выражений на языке С++;

- оператор разветвления вычислений;

- циклические операторы;

- правила объявления арифметических массивов;

- правила обработки и отладки программ обработки арифметических массивов.

***Задание на выполнение лабораторной работы.***

Разработать программу, выполняющую в соответствии со своим вариантом:

1. Вычисление арифметического выражения. Исходные данные представлять как в виде констант, так и в виде переменных. Переменные вводить.

2. Вычисление логического выражения. Исходные данные представлять как в виде констант, так и в виде переменных. Переменные арифметические вводить, логические формировать с использованием арифметических.

3. Вычисление значения функции на основе разложения (в сходящийся ряд) с использованием оператора цикла while.

4. Обработку динамического арифметического массива (Количество строк и столбцов исходного массива вводить с экрана):

- Инициализацию арифметического динамического массива типа double;

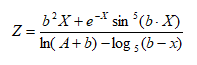
- Вывод арифметического динамического массива (использовать функцию);

- Обработку динамического массива и формирование нового динамического массива (использовать функции с параметрами, возвращающими результат);

- Вывод результатов обработки, в том числе исходного массива с внесенными изменениями (использовать функцию).

***Вариант №2***

***Задание 1:***



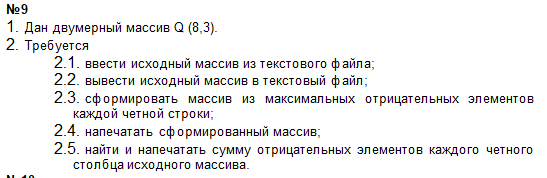
***Задание 2:***



***Задание 3:***



***Задание 4:***

Массив:



***Код программы***

#include <iostream>

#include <cmath>

#include <cstdlib>

#include <fstream>

#include <cstdio>

#include <string>

using namespace std;

// FIRST TASK

double MyFunc(double x);

void CalcFunc();

#define A 1

#define B 10e10

// SECOND TASK

bool MyLogicFunc(bool a, bool b, bool c);

void CalcLogicFunc();

// THIRD TASK

int Factorial(int n);

double FuncForSeq(int n, double x);

void CalcSeq();

#define EPS 1e-5

// FOURTH TASK

int GetRowCount();

int GetColCount();

double\* GetArr(int m, int n);

double\* GetArrAftProc(double arr[], int m, int n);

void PrintSumArr(double arr[], int m, int n);

void InputArr(char filename[], double arr[], int m, int n);

void OutputArr(double arr[], int m, int n);

void OutputArrToFile(char filename[], double arr[], int m, int n);

void ArrTask();

#define M 8

#define N 3

int main()

{

setlocale(LC\_ALL, "Russian");

int task;

bool work = true;

char branch;

do {

cout << "\nВыберите задачу: \n";

cout << "1 - Вычисление арифметического выражения\n";

cout << "2 - Вычисление логического выражения\n";

cout << "3 - Вычисление суммы ряда функции\n";

cout << "4 - Работа с массивом\n";

cout << "Любая другая клавиша чтобы выйти\n";

cin >> branch;

system("cls");

switch (branch)

{

case '1': CalcFunc();

break;

case '2': CalcLogicFunc();

break;

case '3': CalcSeq();

break;

case '4': ArrTask();

break;

default: work = false;

}

} while (work);

return 0;

}

// FIRST TASK

double MyFunc(double x)

{

return (B \* B \* x + exp(-x) \* pow(sin(B \* x), 5)) / (log(A + B) - log(B - x) / log(5));

}

void CalcFunc()

{

setlocale(LC\_ALL, "RUS");

double x;

cout << "\nВведите x для вычисления функции с A = " << A << "; и B = " << B << endl;

cin >> x;

cout << "Результат F(x) = " << MyFunc(x) << endl;

}

// SECOND TASK

bool MyLogicFunc(bool a, bool b, bool c)

{

return (a && (!b) && (!c)) || ((!a) && b && (!c)) || ((!a) && (!b) && c);

}

void CalcLogicFunc()

{

bool a, b, c;

double x1, x2, x3;

cout << "Введите число для сравнения X > 0\n\n";

cin >> x1;

cout << "Введите число для сравнения X < 0\n\n";

cin >> x2;

cout << "Введите число для сравнения X = 0\n\n";

cin >> x3;

a = x1 > 0;

b = x2 < 0;

c = x3 == 0;

cout << "Результат логического выражения на основе трех сравнений, где истина, когда только одна переменная - истина, а остальное - ложь:\n"

<< (MyLogicFunc(a, b, c) ? "True\n" : "False\n");

}

// THIRD TASK

int Factorial(int n)

{

int res = 1;

if (n >= 2)

{

for (int i = 2; i <= n; i++)

{

res \*= i;

}

}

return res;

}

double FuncForSeq(int n, double x)

{

int one = ((n % 2) ? 1 : -1);

return one \* pow(2, 2 \* n - 1) \* pow(x, 2 \* n) / Factorial(2 \* n);

}

void CalcSeq()

{

double sum = 0;

double x;

int n = 0;

double res;

cout << "Введите Х для вычисления (sin(x))^2, где |X| < 1;\n";

cin >> x;

if ((x <= -1) || (x >= 1)) {

x = (x <= -1 ? -0.99 : x);

x = (x >= 1 ? 0.99 : x);

cout << "X вне границ сходимости и будет заменено\n";

cout << "на минимальное или минимальное число, при котором ряд сходится\n";

}

do

{

n++;

res = FuncForSeq(n, x);

sum += res;

} while (fabs(res) > EPS);

cout << "X = " << x << endl;

cout << "Сумма ряда Тейлора функции\n";

cout << sum << endl;

cout << "\nМатематическая функция\n"

<< sin(x) \* sin(x) << endl;

cout << "\nМатематическая функция sin(x)\n"

<< sin(x) << endl;

}

// FOURTH TASK

int GetRowCount()

{

int m;

cout << "Введите число строк\n";

cin >> m;

system("cls");

return m;

}

int GetColCount()

{

int n;

cout << "Введите число столбцов\n";

cin >> n;

system("cls");

return n;

}

void OutputArr(double arr[], int m, int n)

{

string s = string(11 \* n + 1, '=');

cout << s << endl;

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

{

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

{

printf("|%10.3f", arr[i \* n + j]);

}

cout << "|" << endl;

cout << s << endl;

}

}

double\* GetArr(int m, int n)

{

double\* arr = new double[m \* n];

return arr;

}

void InputArr(char filename[], double arr[], int m, int n)

{

ifstream fin(filename);

if (!fin)

cout << "Error!" << endl;

else

{

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

{

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

{

fin >> arr[i \* n + j];

}

}

}

fin.close();

}

double\* GetArrAftProc(double arr[], int m, int n)

{

int l\_m = m / 2;

double\* minmax = new double[l\_m];

for (int i = 0; i < l\_m; i++)

{

double max = -1e6;

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

{

if (arr[(i \* 2 + 1) \* n + j] < 0)

{

max = ((arr[(i \* 2 + 1) \* n + j] > max) ? arr[(i \* 2 + 1) \* n + j] : max);

}

}

minmax[i] = max;

}

return minmax;

}

void PrintSumArr(double arr[], int m, int n)

{

int l\_n = n / 2;

for (int j = 0; j < l\_n; j++)

{

double sum = 0;

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

{

if (arr[i \* n + j \* 2 + 1] < 0)

{

sum += arr[i \* n + j \* 2 + 1];

}

}

cout << "\t" << sum << endl;

}

}

void ArrTask() {

bool deny;

int m;

int n;

do {

deny = false;

cout << "Введите размер массива, где количество элементов не превышает 24\n";

m = GetRowCount();

n = GetColCount();

system("cls");

if ((m \* n) > 24 || (m < 0) || (n < 0) )

{

cout << "Количество элементов массива некорректно, повторите ввод\n";

deny = true;

}

} while (deny);

int minmax\_m = m / 2;

char filename[] = "nums.txt";

double\* minmax;

double\* arr = GetArr(m, n);

InputArr(filename, arr, m, n);

minmax = GetArrAftProc(arr, m, n);

cout << "Исходный массив" << endl;

OutputArr(arr, m, n);

cout << endl;

cout << "Массив максимальных отрицательных чисел каждой четной строки:\n";

OutputArr(minmax, minmax\_m, 1);

cout << "\nСумма отрицательных элементов для четных столбцов:\n";

PrintSumArr(arr, m, n);

\*filename = \*("output.txt");

OutputArrToFile(filename, arr, m, n);

delete[] minmax;

delete[] arr;

}

void OutputArrToFile(char filename[], double arr[], int m, int n) {

ofstream fout(filename);

if (!fout)

cout << "Error!" << endl;

else

{

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

{

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

{

fout << arr[i \* n + j] << ' ';

}

fout << endl;

}

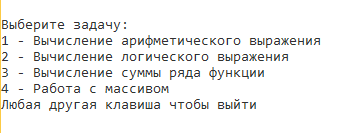
}

fout.close();

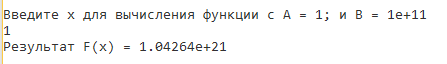
}

***Результаты работы программы***

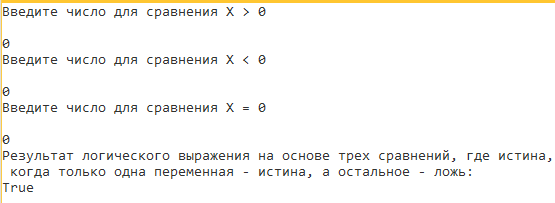
При запуске:

******

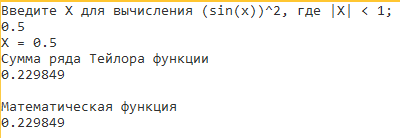
При нажатии «1»:

******

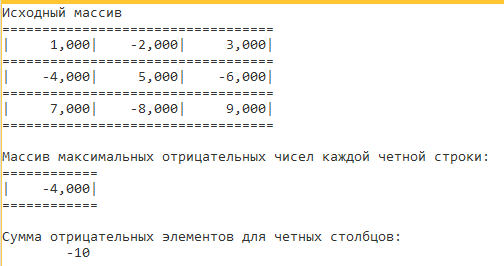
При нажатии «2»:

******

При нажатии «3»:

******

При нажатии «4»:

******

Выведенный в файл массив:

******