**Министерство науки и высшего образования РФ**

**ФГБОУ ВО**

**Московский авиационный институт**

**(национальный исследовательский университет)**

Институт №3.

Системы управления, информатика и электроэнергетика.

Кафедра 304.

**Отчет по лабораторной работе №14**

**по учебной дисциплине  
 «Программирование на языках высокого уровня»  
на тему  
«Закрепление пройденного материала: основные элементы языка Си, массивы, условные операторы, функции, стандартные библиотеки»**

Выполнили студенты группы М3О-224Б-18:

Мариненко Г. В.

Рогачев В.Н.

Приняли:

Доцент кафедры 304, к. т. н. Новиков П.В.\_\_\_\_\_\_

Доцент, к. т. н. Чебатко М.И.\_\_\_\_\_\_

Москва 2019

Задание:

2 вариант:

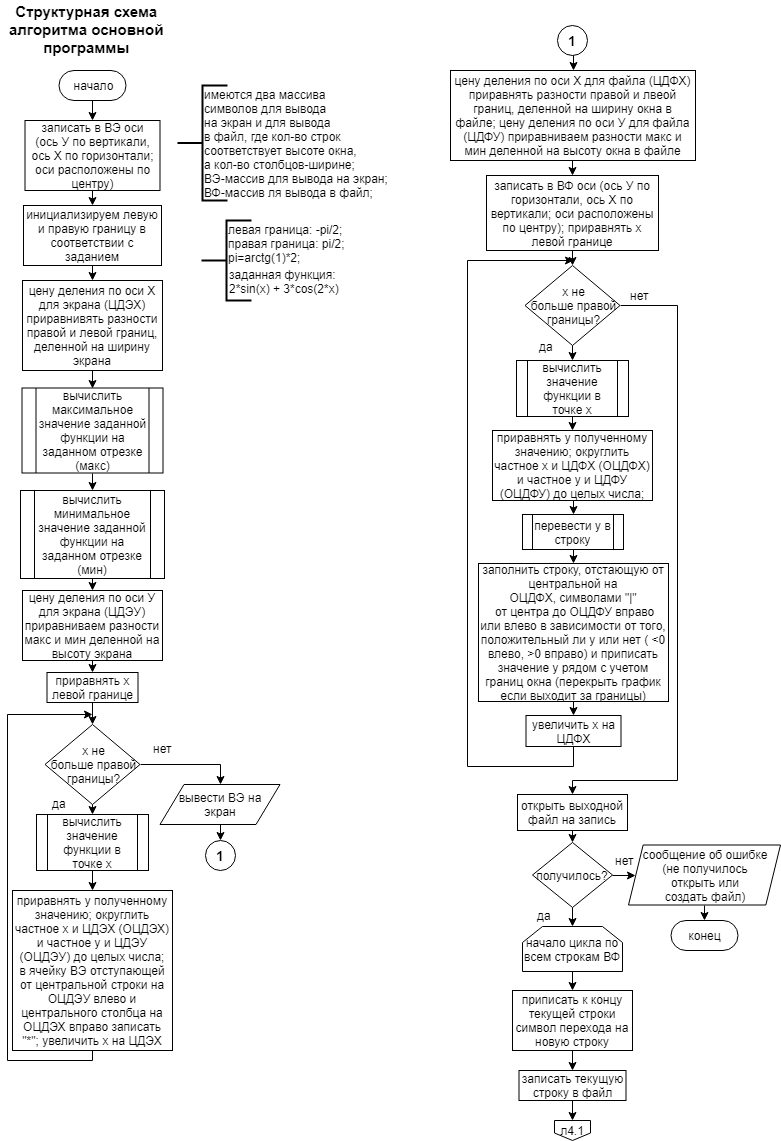
Разработать программу и подпрограмму (подпрограммы) построения графика функции на экране в виде последовательности точек и в выходном файле в виде гистограммы.

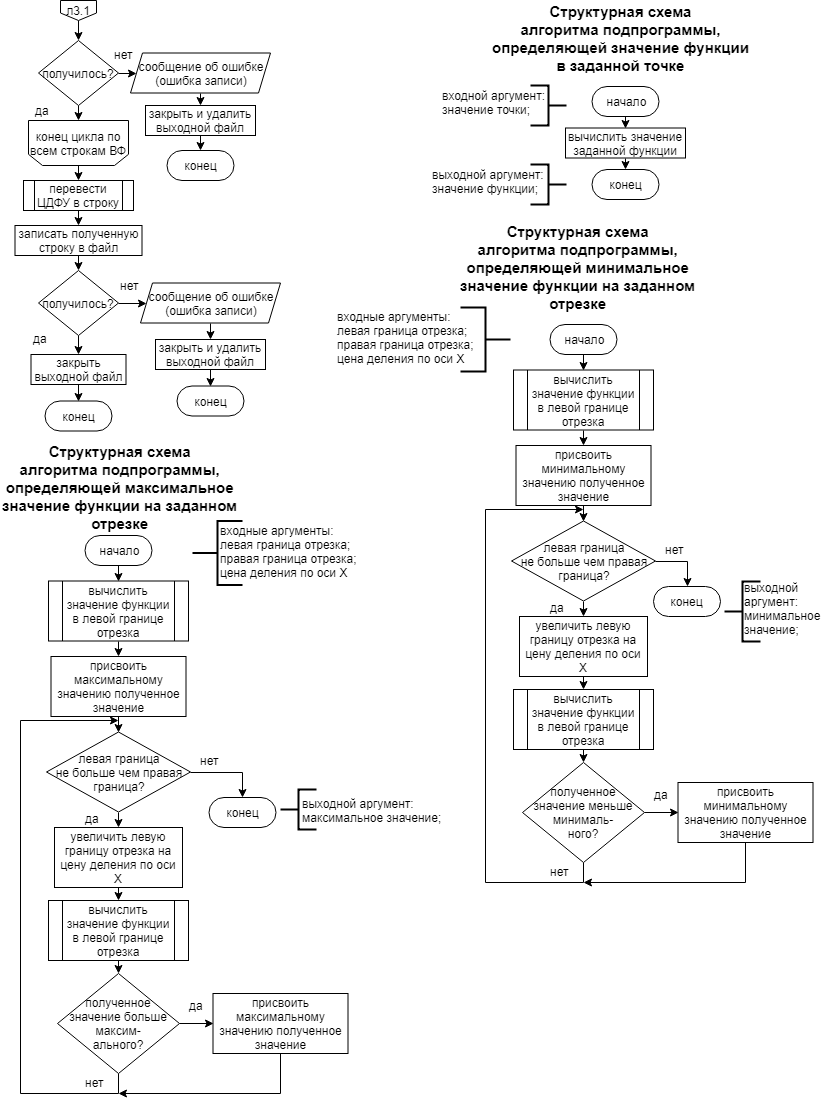
При выводе графика функции на экран ось X расположить горизонтально (количество позиций по оси X = 78), а ось Y – вертикально (количество позиций по оси Y = 25).

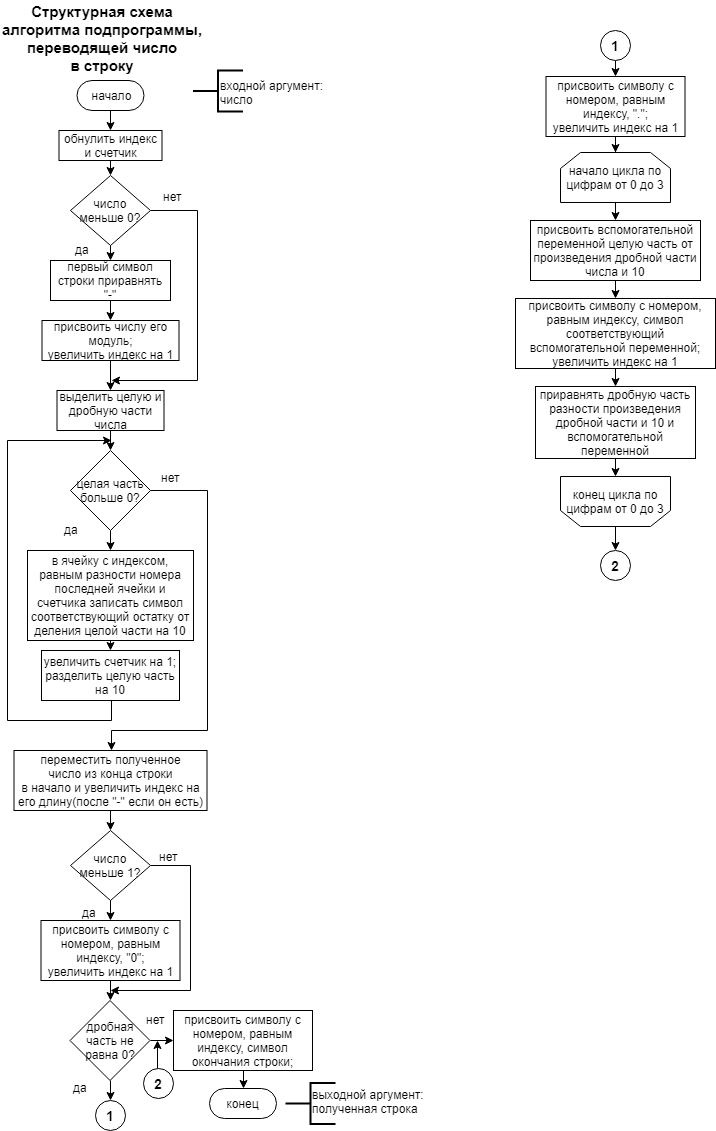
При выводе графика функции в выходной файл ось X расположить вертикально (количество позиций по оси X = 60), а ось Y – горизонтально (количество позиций по оси Y = 80), каждый столбец гистограммы пометить значением функции. После гистограммы вывести масштаб функции по оси Y.



Структурная схема алгоритма программы:







Текст программы:

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

\*ProjectName : Lab7 \*

\*Project.Type: Win32 Console Application \*

\*File Name : Lab7.cpp \*

\*Language : C \*

\*Programer : Рогачев Владислав Николаевич, \*

\* Мариненко Георгий Владимирович \*

\*М3О-224Б-18 Бригада №8 \*

\*Created : 29/11/19 \*

\*LastRevision: 08/12/19 \*

\*Comment : Закрепление пройденного материала: основные элементы языка Си,\*

массивы, условные операторы, функции, стандартные библиотеки. \*

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

#include "pch.h"

#include "string.h"

#include "math.h"

#include "stdio.h"

#include "stdlib.h"

#pragma warning(disable : 4996)

double Function(double x);

double FindMax(double a, double b, double step);

double FindMin(double a, double b, double step);

char \*FromDoubleToString(double x);

int main()

{

char filename[256] = { '\0' };

strcat(filename, "output.txt");

char console[30][84];

int i;

int j;

for (i = 0; i < 30; i++)

{

for (j = 0; j < 83; j++)

{

console[i][j] = ' ';

}

}

for (i = 0; i < 30; i++)

{

console[i][83] = '\0';

}

for (i = 0; i < 30; i++)

{

console[i][41] = (char)179;

}

for (i = 0; i < 83; i++)

{

console[12][i] = (char)196;

}

console[12][41] = (char)197;

double a = -1 \* (2 \* atan(1));

double b = 2 \* atan(1);

double valueXconsole = (b - a) / 78;

double valueYconsole = (FindMax(a, b, valueXconsole) - FindMin(a, b, valueXconsole)) / 25;

double x = a;

while (x <= b)

{

double Y = Function(x);

int countXconsole = round(x / valueXconsole);

int countYconsole = round(Y / valueYconsole);

console[12 - countYconsole][41 + countXconsole] = '\*';

x += valueXconsole;

}

for (i = 0; i < 25; i++)

{

printf("%s\n", console[i]);

}

char file[65][96];

double valueXfile = (b - a) / 60;

double valueYfile = (FindMax(a, b, valueXfile) - FindMin(a, b, valueXfile)) / 80;

for (int i = 0; i < 65; i++)

{

for (j = 0; j < 85; j++)

{

file[i][j] = ' ';

}

}

for (i = 0; i < 65; i++)

{

file[i][85] = '\0';

}

for (i = 0; i < 65; i++)

{

file[i][50] = '|';

}

for (i = 0; i < 85; i++)

{

file[30][i] = '\_';

}

file[30][50] = (char)197;

x = a;

while (x <= b)

{

double Y = Function(x);

int countXfile = round(x / valueXfile);

int countYfile = round(Y / valueYfile);

char \*s = FromDoubleToString(Y);

if (countYfile > 0)

{

for (i = 0; i <= countYfile; i++)

file[30 - countXfile][50 + i] = '|';

if ((50 + countYfile + strlen(s)) < 95)

{

int j = 0;

for (i = countYfile; i < countYfile + 1 + strlen(s); i++)

{

file[30 - countXfile][50 + i] = s[j];

j++;

}

}

else

{

for (i = 0; i < strlen(s); i++)

{

file[30 - countXfile][94 - strlen(s) + i] = s[i];

}

}

}

else

{

for (i = 0; i >= countYfile; i--)

file[30 - countXfile][50 + i] = '|';

int l = (50 + countYfile - strlen(s));

if (l > 0)

{

int j = 0;

for (i = countYfile; i > countYfile - strlen(s); i--)

{

file[30 - countXfile][50 + i] = s[strlen(s) - 1 - j];

j++;

}

}

else

{

for (i = 0; i < strlen(s); i++)

{

file[30 - countXfile][i] = s[i];

}

}

}

x += valueXfile;

}

FILE \*on;

on = fopen(filename, "wt");

if (on == NULL)

{

printf("file not found or can't be created\n");

return 0;

}

for (i = 0; i < 65; i++)

{

strcat(file[i], "\n");

if (fputs(file[i], on) == EOF)

{

fclose(on);

printf("error of writing\n");

remove(filename);

return 0;

}

}

char \*s = FromDoubleToString(valueYfile);

int m = strlen(s);

char temp1[256] = { 0 };

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

{

temp1[j] = s[j];

}

if (fputs(temp1, on) == EOF)

{

fclose(on);

printf("error of writing\n");

remove(filename);

return 0;

}

fclose(on);

getchar();

return 0;

}

double Function(double x)

{

return 2 \* sin(x) + 3 \* cos(2 \* x);

}

double FindMax(double a, double b, double step)

{

double max = Function(a);

while (a <= b)

{

a += step;

if (Function(a) > max)

max = Function(a);

}

return max;

}

double FindMin(double a, double b, double step)

{

double min = Function(a);

while (a <= b)

{

a += step;

if (Function(a) < min)

min = Function(a);

}

return min;

}

char \*FromDoubleToString(double x)

{

char s[10];

int i = 0;

int j = 0;

int count = 0;

if (x < 0)

{

s[0] = '-';

i++;

x = fabs(x);

}

int n = x;

double r = x - n;

while (n > 0)

{

int l = fmod(n, 10);

s[sizeof(s) - count - 1] = (char)l + 48;

count++;

n /= 10;

}

for (j = count; j > 0; j--)

{

s[i] = s[sizeof(s) - j];

i++;

}

if (x < 1)

{

s[i] = '0';

i++;

}

if (r != 0)

{

s[i] = '.';

i++;

for (j = 0; j < 3; j++)

{

int a = r \* 10;

s[i] = (char)a + 48;

r = r \* 10 - a;

i++;

}

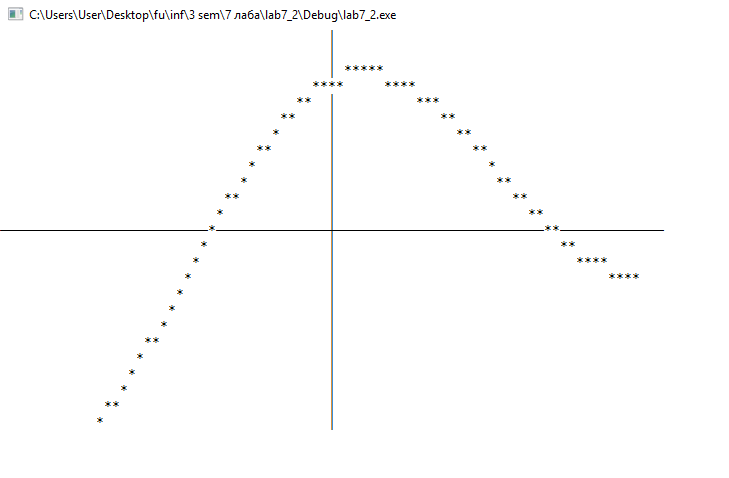
}

s[i] = '\0';

return s;

}

Результат работы программы:



Содержимое выходного файла:

