**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 «Закрепление пройденного материала: основные элементы языка Си, массивы, условные операторы, функции, стандартные библиотеки»**

**на тему**

***«Изучение принципов работы с текстовыми файлами»***

Группа: *М3О-209Б-19*

*Вариант №4*

Выполнил: студент группы М3о-209б-19 *Кузнецов И.И.*

Приняли: **Доцент к. 304, к.т.н., Новиков П. В.**

**Старший преподаватель к. 304, Ивашенцев И. В.**

Москва 2020

**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 «Закрепление пройденного материала: основные элементы языка Си, массивы, условные операторы, функции, стандартные библиотеки»**

**на тему**

***«Изучение принципов работы с текстовыми файлами»***

Группа: *М3О-209Б -19*

*Вариант №4*

Выполнил: студент группы М3о-209б-19 *Мухаметгалиев А.Р.*

Приняли: **Доцент к. 304, к.т.н., Новиков П. В.**

**Старший преподаватель к. 304, Ивашенцев И. В.**

Москва 2020

**Содержание**

1. **Задание.**
2. **Структурная схема алгоритма программы и подпрограмм.**
3. **Текст программы.**
4. **Результаты работы программы.**
5. **Вывод.**

**Задание**

**Вариант 4**

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

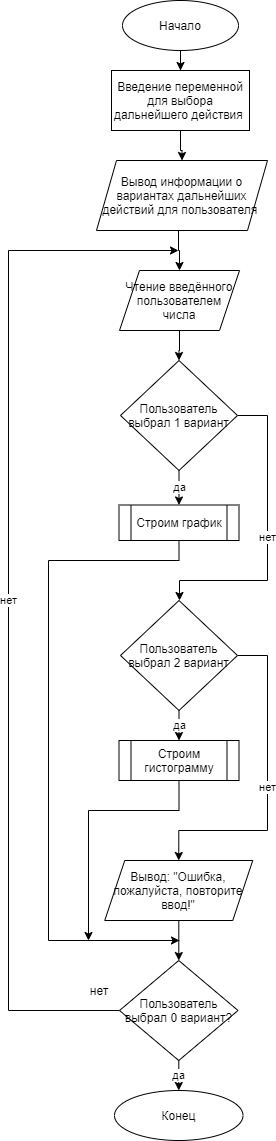
При выводе графика функции на экран ось X расположить горизонтально (количество позиций по оси X = 78), а ось Y – вертикально (количество позиций по оси Y = 25).

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.

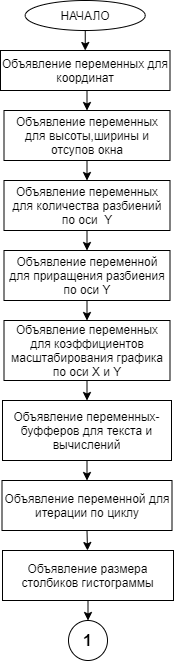
После гистограммы вывести масштаб функции по оси Y. Функция и координаты отрезка приведены в таблице.

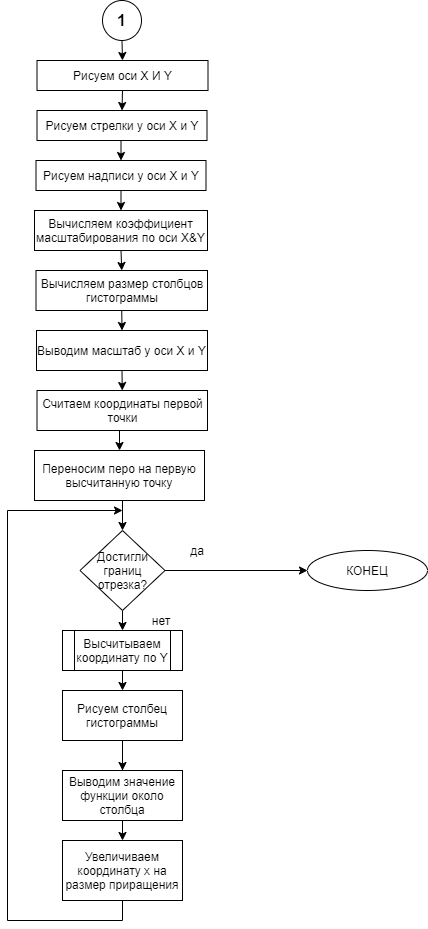
|  |  |  |
| --- | --- | --- |
| Вариант | Функция | Границы отрезка |
| 4 | y = x2 \* exp(-x) | -1 ÷ 2 |

**Структурная схема алгоритма программы.**

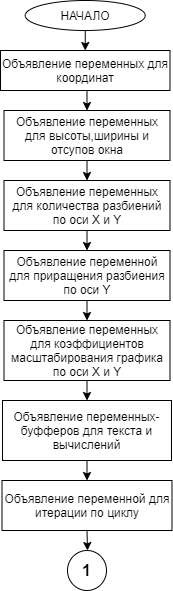


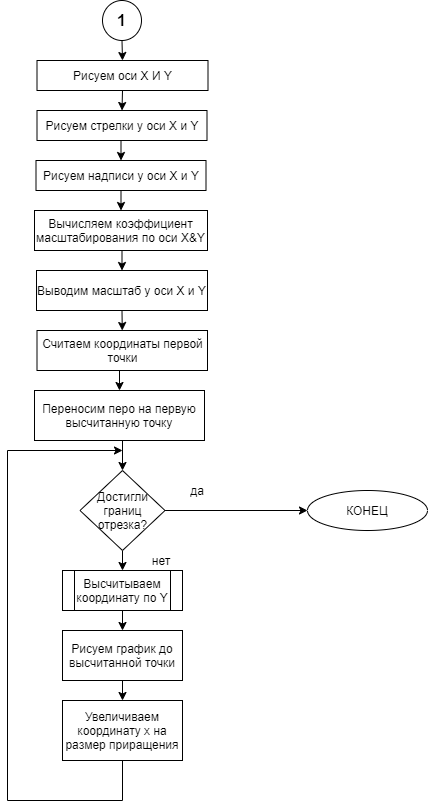
**Структурная схема алгоритма подпрограммы DrawHist.**

****

****

**Структурная схема алгоритма подпрограммы DrawGraf.**

****

****

**Структурная схема алгоритма подпрограммы f.**



**Текст программы**

#include <graphics.h>

#include <conio.h>

#include <math.h>

double f(double x);

void drawGraph();

void drawHist();

int main() {

int choose = -1;

printf("Choose action:\n");

printf("1.Build a graph\n");

printf("2.Build a histogram\n");

printf("0.Exit\n");

do {

scanf("%d", &choose);

if (choose == 1) {

drawGraph();

}

else if (choose == 2) {

drawHist();

}

else if (choose == 0) {

printf("Done\n");

}

else {

printf("Error! Repeat please\n");

}

} while (choose != 0);

return 0;

}

void drawHist() {

double x; //координата x

double y; //координата y

int height = 650; //высота окна

int width = 700; //ширина окна

int startx = 25; //отсуп от начала экрана оси x

int starty = 25; //отсуп от начала экрана оси y

double numberSplitsY; //количество разбиений по оси Y

double dy; //приращение по оси y для рисования масштаба

int coefx; //вычисляем коэффицент масштабирования

int coefy; //вычисляем коэффицент масштабирования

int sizebar; //размер столбиков гистограммы

char temp[4]; //буффер для подписи столбцев гистограммы

double buff; //буффер для вычислений

int i; //итератор по циклу

initwindow(width, height); //инициализируем размеры окна

line(0, height / 3 + startx - 10, width, height / 3 + startx - 10); //рисуем ось x

line(starty, 0, starty, height); //рисуем ось y

setfillstyle(SOLID\_FILL, RED); //устанавливаем стиль столбика

line(width - 10, height / 3 + startx - 20, width, height / 3 + startx - 10);

line(width - 10, height / 3 + startx, width, height / 3 + startx - 10);//стрелка осиy

line(starty - 10, height - 10, starty, height);

line(starty + 10, height - 10, starty, height); //стрелка оси x

outtextxy(width - 10, height / 3 + startx, "Y"); //рисуем надпись у оси Y

outtextxy(starty - 20, height - 15, "X"); //рисуем надпись у оси Х

coefx = (height - startx - 25) / 3; //вычисляем коэффицент масштабирования

coefy = (width - starty \* 4) / 2.718; //вычисляем коэффицент масштабирования

sizebar = (height - startx) / 60 - 1; //вычисляем размер столбиков гистограммы

settextstyle(SMALL\_FONT, HORIZ\_DIR, 0); //устанавливаем стиль шрифтов

//вывод масштаба у оси Y

numberSplitsY = 3;

for (i = 0; i <= numberSplitsY; i++) {

dy = 3 / numberSplitsY;

buff = -1 + dy \* i;

sprintf(temp, "%1.2f", buff);

outtextxy(2, ((height - 50) / numberSplitsY \* i + 27), temp);

line(20, ((height - 50) / numberSplitsY \* i + 31), 30, ((height - 50) / numberSplitsY \* i + 31));

}

//вывод масштаба у оси X

i = 1;

for (i; i < 6; i++) {

y = i / 2.0;

sprintf(temp, "%.3f", y); //из числа в текст

outtextxy(starty + (width - starty) / 6 \* i + 1 + 3, height / 3 + startx - 8, temp); //выводим текст

line(starty + (width - starty) / 6 \* i, height / 3 + startx - 15, starty + (width - starty) / 6 \* i,

height / 3 + startx - 5); //рисуем разделение по оси x

}

x = -1; //начало отрезка

while (x < 2.05) { //конец отрезка

y = f(x);

bar(starty + 1, //левая граница: начало оси ординат + отсуп

x \* coefx + coefx + startx +

1, //верхняя граница: начало оси абсцисс + отсуп + значение аргумента \* коэффицент масттабирования

y \* coefy + starty +

1, //правая граница: начало оси ординат + отсуп + значение функции \* коэффицент масттабирования

x \* coefx + coefx + startx + sizebar + 1 //нижняя граница: аналогично верхней + размер столбика

);

sprintf(temp, "%.3f", y); //переводим значение функции в текстовый формат

outtextxy(y \* coefy + starty + 6, x \* coefx + coefx + startx + 2,

temp); //приписываем около столбика значение функции

x += 0.05; //приращение

};

getch(); // ожидание нажатия пользователем любой клавиши

closegraph(); // выход из графического режима

}

void drawGraph() {

double x; //координата x

double y; //координата y

int height = 325; //высота окна

int width = 600; //ширина окна

int startx = 300; //отсуп от начала экрана оси x

double numberSplitsX; //количество разделений масштаба по оси Х

double numberSplitsY; //количество разделений масштаба по оси Y

double dy; //приращение по оси y для рисования масштаба

int coefx; //коэффицент масштабирования по оси X

int coefy; //коэффицент масштабирования по оси Y

char temp[4]; //буффер для надписей

double buff; //буффер для вычислений

int i = 0; //итератор по циклу

initwindow(width, height); //инициализируем размеры окна

line(0, startx, width, startx); //рисуем ось x

line(width / 3 + 8, 0, width / 3 + 8, height); //рисуем ось y

line(width / 3 - 10 + 8, 15, width / 3 + 8, 0);

line(width / 3 + 10 + 8, 15, width / 3 + 8, 0); //стрелка оси y

line(width - 10, startx - 10, width, startx);

line(width - 10, startx + 10, width, startx); //стрелка оси x

outtextxy(width - 10, startx - 35, "X"); //рисуем надписи у оси Х

outtextxy(width / 3 + 25, 5, "Y"); //рисуем надписи у оси Y

coefx = (width - 50) / 3; //вычисляем коэффицент масштабирования

coefy = (startx - 25) / 2.71;//вычисляем коэффицент масштабирования

//вывод масштаба у оси X

numberSplitsX = 6;

for (i = 0; i <= numberSplitsX; i++) {

buff = 3.0 / numberSplitsX \* i - 1.0;

sprintf(temp, "%1.2f", buff);

line((width - 50) / numberSplitsX \* i + 25, height - 20, (width - 50) / numberSplitsX \* i + 25, height - 30);

outtextxy((width - 50) / numberSplitsX \* i + 22, height - 15, temp);

}

//вывод масштаба у оси Y

numberSplitsY = 5;

for (i = 0; i < numberSplitsY; i++) {

dy = 2.718 / numberSplitsY;

buff = 2.718 - dy \* i;

sprintf(temp, "%1.3f", buff);

outtextxy(width / 3 + 17, ((height - 50) / numberSplitsY \* i + 19), temp);

line(width / 3 + 3, ((height - 50) / numberSplitsY \* i + 25), width / 3 + 13,

((height - 50) / numberSplitsY \* i + 25));

}

x = -1; //начало отрезка

y = f(x); //вычисляем значение функции

moveto(coefx + 25 + x \* coefx, startx - y \* coefy); //переносим перо на начало координат

while (x < 2) { //конец отрезка

y = f(x); //высчитываем y

lineto(coefx + 25 + x \* coefx, startx - y \* coefy); //рисуем график

x += 0.03846; //приращение по оси X 3/78

};

getch(); // ожидание нажатия пользователем любой клавиши

closegraph(); // выход из графического режима

}

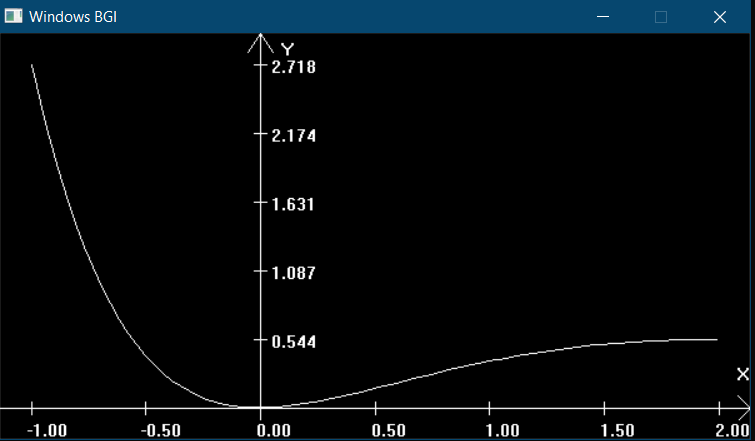
double f(double x) {

return x \* x \* exp(-x);

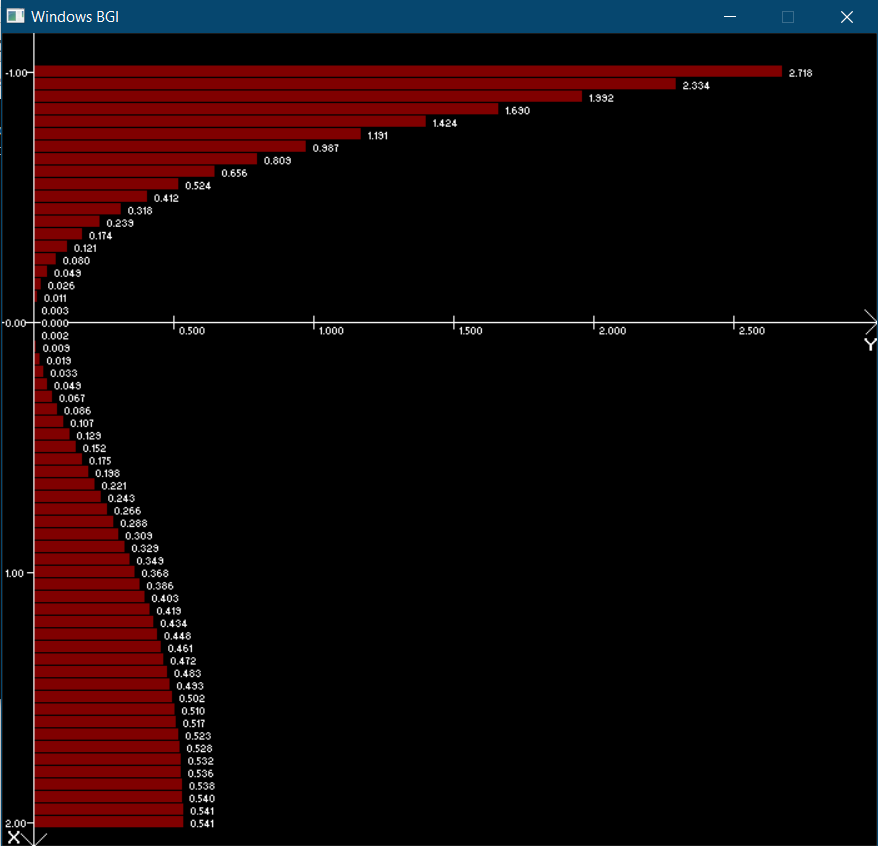
}

**Результаты работы программы**

* **График:**

****

* **Гистограмма**

****

**Вывод**

Мы разработали 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 расположили горизонтально (количество позиций по оси X = 78), а ось Y – вертикально (количество позиций по оси Y = 25). При выводе графика функции на экран ось X расположили 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 расположили вертикально (количество позиций по оси X = 60), а ось Y – горизонтально (количество позиций по оси Y = 80), каждый столбец гистограммы пометили значением функции.

Мы использовали:

-функцию DrawGraf, выполняющую построение графика по заданной функции в заданном интервале,

-функцию DrawHist, выполняющую построение гистограммы по заданной функции в заданном интервале,

-функцию f, вычисляющую значение первой функции в точке х,