# Описание

Программа и подпрограмма построения графика функции на экране в виде последовательности точек и в выходном файле в виде гиcтограммы

Функция:

**Текст программы**

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

\* Project name : - \*

\* Project type : win32 console app \*

\* File name : - \*

\* Programmers : Usupov Tengiz T. \*

\* Modified : - \*

\* Created : 22.11.18 11:10 \*

\* Last Revision : 01.12.18 17:55 \*

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

#include < stdio.h >

#include < glut.h >

#include < math.h >

#pragma warning(disable: 4996)

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

/\* П Р О Т О Т И П Ы Ф У Н К Ц И Й \*/

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

Void BarChart();

float f(float x);

void displayN(float x, float y, int N, bool check);

void GRAPH();

const float x1 = -1, x2 = 4;

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

/\* О С Н О В Н А Я П Р О Г Р А М М А \*/

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

int main(int argc, char \*\* argv) {

BarChart();

glutInit(&argc, argv); //инициализируем glut

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); //использовать 1 буфер, и цвет типа RGB (3 составляющие)

glutInitWindowSize(1200, 600); //зададим размер окна

glutInitWindowPosition(100, 100); //позиция открытия окна

glutCreateWindow("Polygon"); //создать окно

glClearColor(1.0, 1.0, 1.0, 1.0); //цвет фона окна

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-1.5, 5, -2, 3, -1.0, 1.0); //зададим размеры координатной сетки

glutDisplayFunc(GRAPH); //зададим функцию отображающую необходимые объекты

glutMainLoop(); //передает управление в glut

}//main

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

/\* РЕАЛИЗАЦИЯ ФУНКЦИЙ \*/

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

float f(float x)

{

return (x - 3) / (pow(x, 2) + 2);

}//f

void displayN(float x, float y, int N, bool check)

{

glRasterPos2f(x, y);

if (N < 0)

{

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, '-');

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, '0' - N / 10);

}

else

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, '0' + N / 10);

if (check)

{

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, '.');

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, '5');

}

}//displayN

void BarChart() {

FILE \* out;

out = fopen("output.txt", "w");

fclose(out);

out = fopen("output.txt", "a");

if (out)

{

float x, y, tmp, Oy = (f(x2) - f(0.3166248))/40;

for (x = x1; x <= x2; x += (x2 - x1) / 40) // ГИСТОГРАММА

{

y = f(x);

tmp = -1.8;

if (y < 0)

{

while (tmp < y - 8 \* Oy)

{

fprintf(out, " ");

tmp += Oy;

}

fprintf(out, "%.4f ", y);

tmp += 8 \* Oy;

while (tmp < 0)

{

fprintf(out, "@");

tmp += Oy;

}

fprintf(out, "\n");

}

else

{

while (tmp < 0)

{

fprintf(out, " ");

tmp += Oy;

}

tmp = 0;

while (tmp < y)

{

fprintf(out, "@");

tmp += Oy;

}

fprintf(out, " %.4f\n", y);

}

}

fprintf(out, "\n\nMasshtab po osi Y: %f", Oy);

fclose(out);

}

else

{

printf("Error");

return;

}

}//BarChart

void GRAPH()

{

glClear(GL\_COLOR\_BUFFER\_BIT); //очистим буфер

glColor3f(0.0, 0.0, 0.0); //цвет

glBegin(GL\_LINES); //ось OX

glVertex2f(-1.4, 0);

glVertex2f(4.4, 0);

glVertex2f(4.4, 0);

glVertex2f(4.3, 0.03);

glVertex2f(4.4, 0);

glVertex2f(4.3, -0.03);

glEnd();

glRasterPos2f(4.32, 0.05);

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, 'X');

glBegin(GL\_LINES); //ось OY

glVertex2f(0, 2.9);

glVertex2f(0, -1.9);

glVertex2f(0, 2.9);

glVertex2f(0.03, 2.8);

glVertex2f(0, 2.9);

glVertex2f(-0.03, 2.8);

glEnd();

glRasterPos2f(0.05, 2.83);

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, 'Y');

for (float x = -10, y = -10; x <= 40; x += 5, y += 5) // вставляем значения на осях

{

bool check = 0;

int tmp = x;

if (tmp % 10 != 0) check = 1;

displayN(x / 10, 0.05, x, check);

glBegin(GL\_LINES);

glVertex2f(x / 10, 0.03);

glVertex2f(x / 10, -0.03);

glEnd();

if (y != 0)

{

tmp = y;

if (tmp % 10 != 0) check = 1;

displayN(0.05, y / 10, y, check);

glBegin(GL\_LINES);

glVertex2f(0.03, y / 10);

glVertex2f(-0.03, y / 10);

glEnd();

}

}

glPointSize(4.0); // вставляем точки

glBegin(GL\_POINTS);

glColor3f(1.0, 0.0, 0.0); //выберем для неё красный цвет

for (float x = x1; x <= x2; x += (x2 - x1) / 100)

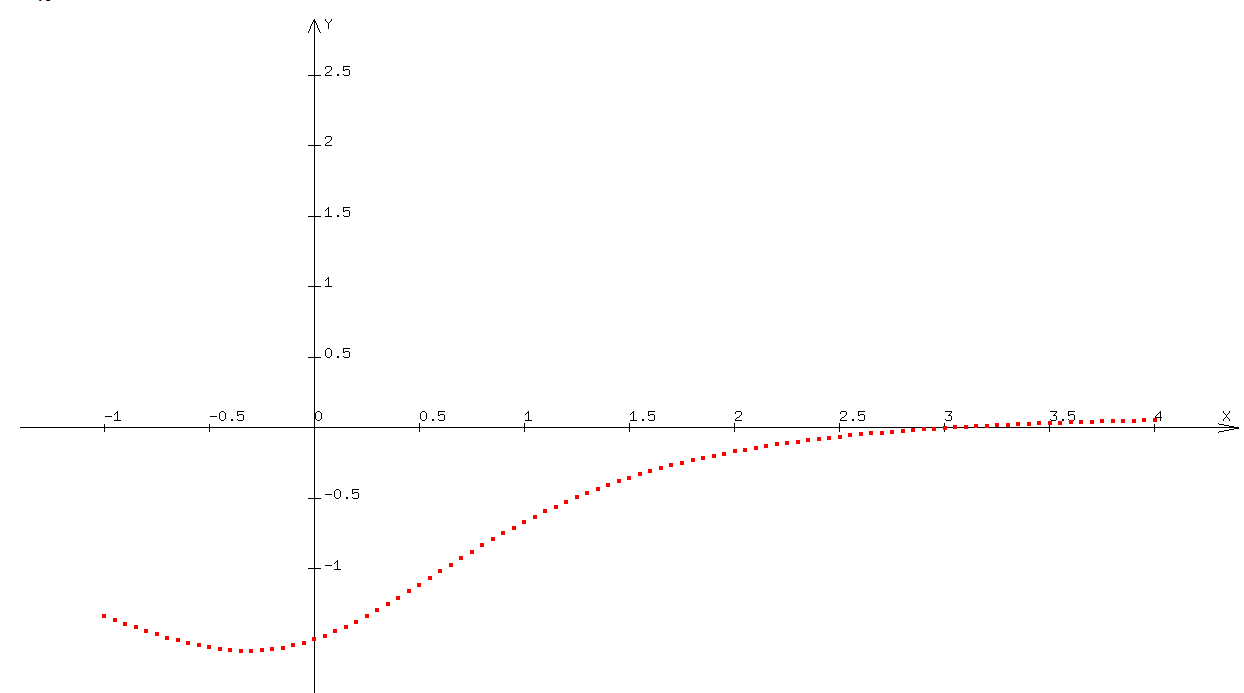
{

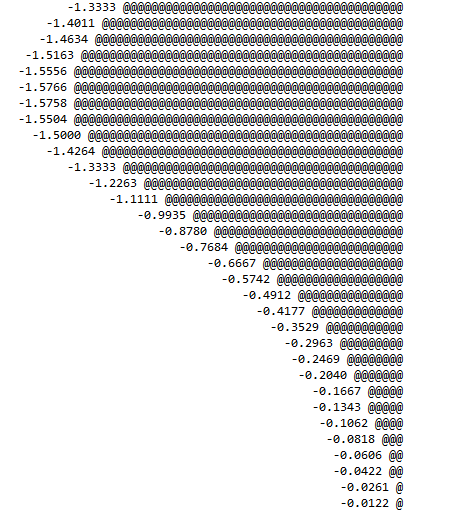
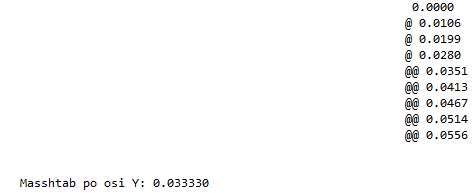
glVertex2f(x, f(x));

}

glEnd();

glFlush();

}//GRAPH **Результаты работы программы**

****