# Самостоятельная работа

## Задание 1

Нарисовать график функции y=x\*x

## Код программы

#include <graphics.h>

#include <stdio.h>

int main() {

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

int xmin = -3, xmax = 3, ymin = -9, ymax = 9, midx, midy;

float xscale, yscale, x;

midx = getmaxx()/2; midy = getmaxy()/2;

xscale=getmaxx()/(xmax-xmin), yscale=getmaxy()/(ymax-ymin);

setcolor(WHITE);

setfillstyle(SOLID\_FILL, WHITE);

rectangle(0,0,getmaxx(),getmaxy());

floodfill(1,1,WHITE);

setcolor(BLACK);

line(0,getmaxy()/2,getmaxx(),getmaxy()/2);

line(getmaxx()/2,0,getmaxx()/2,getmaxy());

setcolor(BLUE);

x=xmin;

moveto(x\*xscale\*midx, midy-x\*x\*yscale);

while (x<=xmax) {

lineto(x\*xscale+midx,midy-x\*x\*yscale);

x+=0.1;

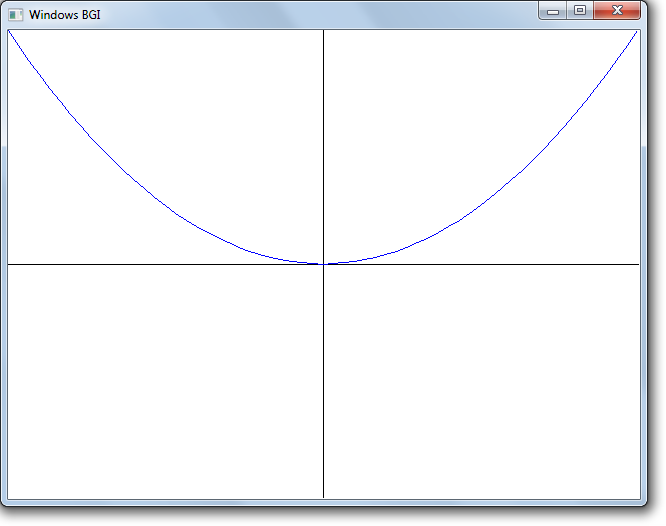
}

getch();

return 0;

}

## Результат проделанной работы



## Задание 2

Написать программу, которая использует стандартные

функции для рисования на белом фоне красного прямоугольника с синей

границей и синими диагоналями, и желтого круга с фиолетовой границей

в центре экрана.

## Код программы

#include <graphics.h>

#include <stdio.h>

int main() {

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

setcolor(WHITE);

setfillstyle(SOLID\_FILL, WHITE);

rectangle(0,0,getmaxx(),getmaxy());

floodfill(1,1,WHITE);

setcolor(BLUE);

setfillstyle(SOLID\_FILL, RED);

rectangle(1,1,150,75);

floodfill(11,11,BLUE);

line(1,1,150,75);

line(150,1,1,75);

setcolor(MAGENTA);

setfillstyle(SOLID\_FILL, YELLOW);

ellipse(getmaxx()/2,getmaxy()/2,0,360,180,180);

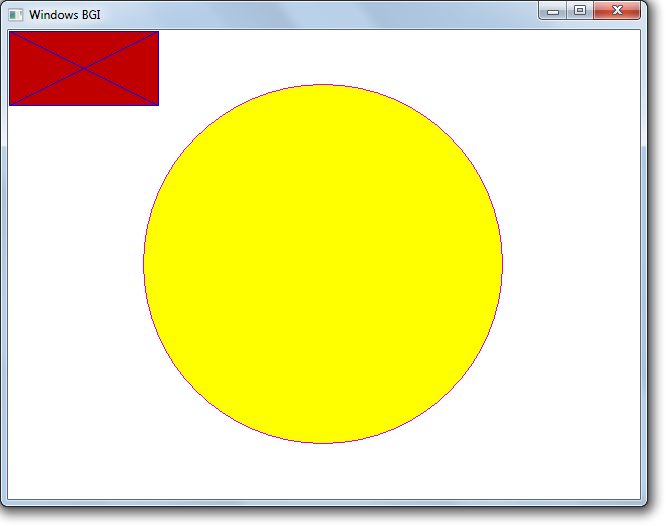
fillellipse(getmaxx()/2,getmaxy()/2,180,180);

getch();

return 0;

}

## Результат проделанной работы



## Задание 3

Построить график циклоиды.

Математическое описание:

x = a t - b sin t, y = a - b cos t,

при b<a циклоида получается укороченной, при b>a - удлиненной, b = a

дает обычную циклоиду.

## Код программы

#include <graphics.h>

#include <stdio.h>

#include <math.h>

#include<locale.h>

int main() {

char \*locale = setlocale(LC\_ALL, "");

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

float a,b, xscale=5, yscale, t=0, x, y;

int midx, midy;

printf("Введите a: ");

scanf("%f",&a);

printf("Введите b: ");

scanf("%f",&b);

yscale = getmaxy()/(10\*b);

midx=0; midy=getmaxy();

midx = getmaxx()/2, midy = getmaxy()/2;

x=a\*t-b\*sin(t);

y=a-b\*cos(t);

moveto(x\*xscale+midx,midy-y\*yscale);

setcolor(WHITE);

while (t<50) {

x=a\*t-b\*sin(t);

y=a-b\*cos(t);

lineto(x\*xscale+midx,midy-y\*yscale);

t+=0.1;

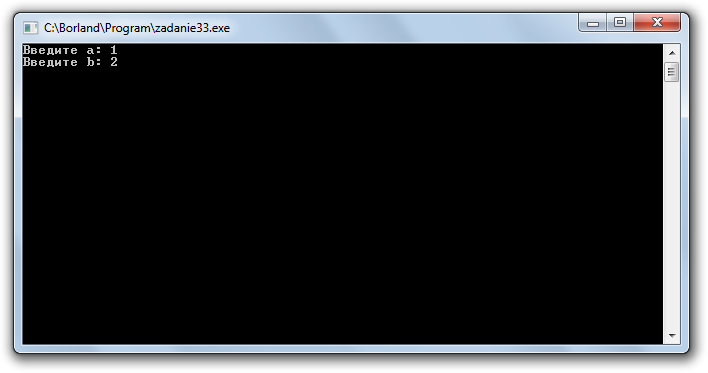
}

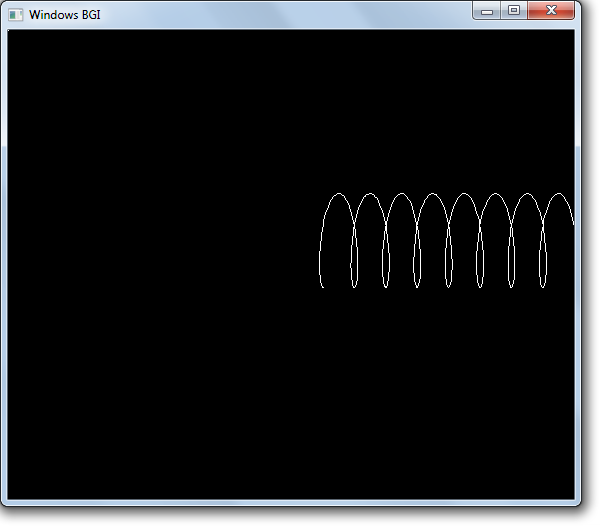
getch();

return 0;

}

## Результат проделанной работы





## Задание 4

Подготовить и отладить программу вычерчивания квадрата,

вписанного в окружность. Запрограммировать изменение цветов

окружности.

## Код программы

#include <graphics.h>

#include <stdio.h>

#include <math.h>

#include<locale.h>

int main() {

char \*locale = setlocale(LC\_ALL, "");

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

float r,scale=1, x1,y1,x2,y2;

int color, midx,midy, xs1,ys1,xs2,ys2;

printf("Введите радиус окружности, описываемой квадрат: ");

scanf("%f",&r);

printf("Нажимайте пробел для смены цвета");

midx = getmaxx()/2;

midy = getmaxy()/2;

x1 = r\*cos(M\_PI/4);

y1 = r\*sin(M\_PI/4);

x2 = r\*cos(M\_PI+M\_PI/4);

y2 = r\*sin(M\_PI+M\_PI/4);

xs1 = midx+x1\*scale;

ys1 = midy+y1\*scale;

xs2 = midx+x2\*scale;

ys2 = midy+y2\*scale;

setcolor(WHITE);

rectangle(xs1,ys1,xs2,ys2);

for (color=0; color <=15; color++) {

setcolor(color);

ellipse(midx,midy,0,360,r,r);

getch();

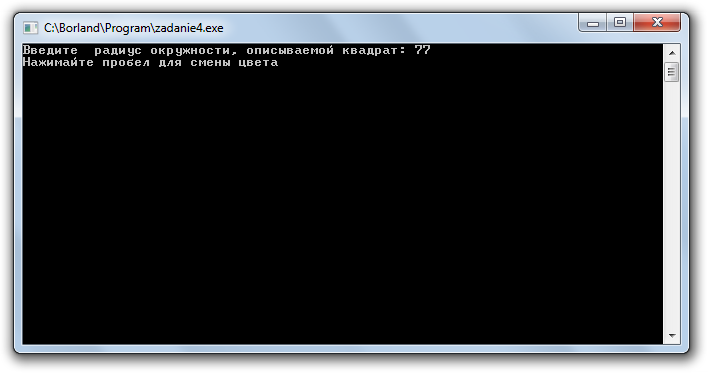
}

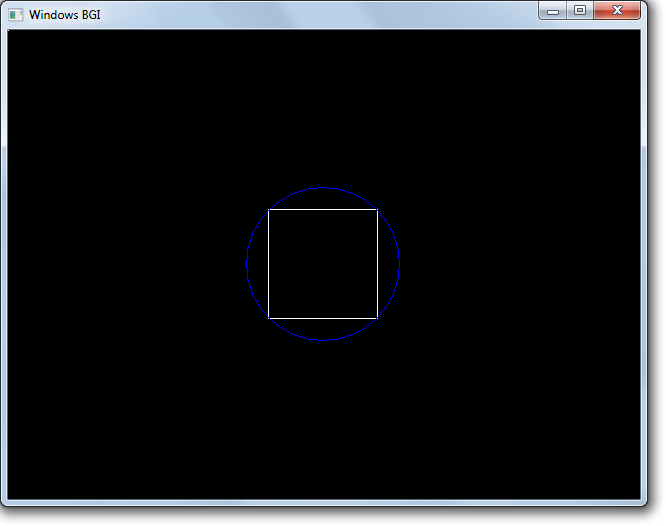
getch();

return 0;

}

## Результат проделанной работы





## Задание 5

Разработать программу изображения точки, движущейся по

косинусоиде

(y = cos(x)).

## Код программы

#include <graphics.h>

#include <stdio.h>

#include <math.h>

#include <time.h>

void anim(float tochka) {

cleardevice();

float yscale, xscale, x=0;

int midx, midy;

yscale = getmaxy()/3;

xscale = getmaxx()/10;

midx = 0;

midy = getmaxy()/2;

setcolor(WHITE);

moveto(x\*xscale+midx,cos(x)\*yscale+midy);

setfillstyle(SOLID\_FILL, WHITE);

while (x<10) {

if (x==tochka) {

fillellipse(x\*xscale+midx,cos(x)\*yscale+midy,5,5);

}

float y = cos(x);

lineto(x\*xscale+midx,cos(x)\*yscale+midy);

x+=0.1;

}

}

int main() {

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

float tochka = 0;

while (tochka <10) {

anim(tochka);

Sleep(10);

tochka+=0.1;

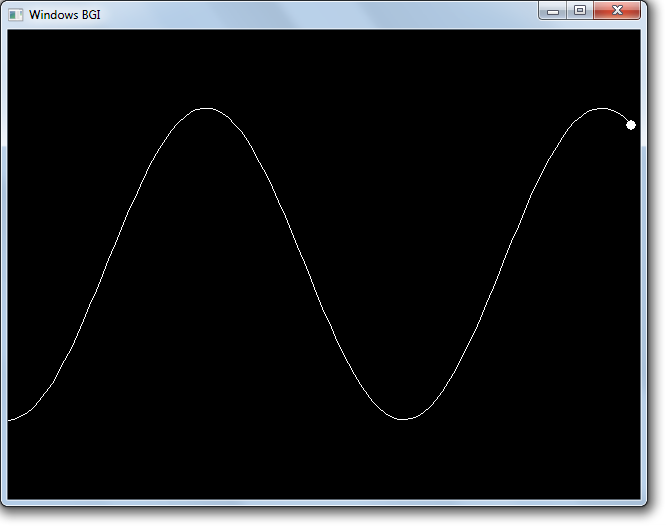
}

getch();

return 0;

}

## Результат проделанной работы



## Задание 6

Изобразить движущийся на зрителя экран (прямоугольник).

## Код программы

#include <graphics.h>

#include <stdio.h>

#include <math.h>

void rectangle(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int y4) {

moveto(x1,y1);

lineto(x2,y2);

lineto(x3,y3);

lineto(x4,y4);

lineto(x1,y1);

}

int main() {

int gd = DETECT, gm; initgraph(&gd, &gm, 0);

int sizex = 600, sizey = 600, midx, midy, x1,x2,x3,x4,y1,y2,y3,y4,xn1,xn2,xn3,xn4,yn1,yn2,yn3,yn4;

float period = 0.2;

midx = getmaxx()/2;

midy = getmaxy()/2;

x1 = midx-sizex/2;

y1 = midy-sizey/2;

x2 = midx+sizex/2;

y2 = midy-sizey/2;

x3 = midx+sizex/2;

y3 = midy+sizey/2;

x4 = midx-sizex/2;

y4 = midy+sizey/2;

setcolor(RED);

for (int i=0;i<20;i++) {

rectangle(x1,y1,x2,y2,x3,y3,x4,y4);

xn1 = x1+(x2-x1)\*period;

yn1 = y1+(y2-y1)\*period;

xn2 = x2+(x3-x2)\*period;

yn2 = y2+(y3-y2)\*period;

xn3 = x3+(x4-x3)\*period;

yn3 = y3+(y4-y3)\*period;

xn4 = x4+(x1-x4)\*period;

yn4 = y4+(y1-y4)\*period;

x1=xn1;

y1=yn1;

x2=xn2;

y2=yn2;

x3=xn3;

y3=yn3;

x4=xn4;

y4=yn4;

}

getch();

return 0;

}

## Результат проделанной работы

