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:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | #include "graphics.h"  #include "stdio.h"  **int** **main**() {  **int** gd, gm;  gd = DETECT;  //graphics mode  initgraph(&gd, &gm, "");  **int** errorcode = graphresult();  //in case of error - print and exit  **if** (errorcode!=grOk) {  printf("Error: %s",grapherrormsg(errorcode));  **return** **1**;  }  //draw here    //white background  setcolor(WHITE);  setfillstyle(SOLID\_FILL, WHITE);  rectangle(**0**,**0**,getmaxx(),getmaxy());  floodfill(**1**,**1**,WHITE);    //blue border with red fill  setcolor(BLUE);  setfillstyle(SOLID\_FILL, RED);  rectangle(**10**,**10**,**200**,**100**);  floodfill(**11**,**11**,BLUE);  //diagonals  line(**10**,**10**,**200**,**100**);  line(**200**,**10**,**10**,**100**);    setcolor(MAGENTA);  setfillstyle(SOLID\_FILL, YELLOW);  ellipse(getmaxx()/**2**,getmaxy()/**2**,**0**,**360**,**100**,**100**);  fillellipse(getmaxx()/**2**,getmaxy()/**2**,**100**,**100**);    //wait for key and exit  getch();  **return** **0**;  } |

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:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | #include "graphics.h"  #include "stdio.h"  #include "math.h"  **int** **main**() {  **int** gd, gm;  gd = DETECT;  //graphics mode  initgraph(&gd, &gm, "");  **int** errorcode = graphresult();  //in case of error - print and exit  **if** (errorcode!=grOk) {  printf("Error: %s",grapherrormsg(errorcode));  **return** **1**;  }    **float** a;  **float** b;  printf("Enter a: ");  scanf("%f",&a);  printf("Enter b: ");  scanf("%f",&b);    //scale  **float** xscale = **5**;  **float** yscale = getmaxy()/(**6**\*b);    //position of center  **int** x0 = **0**;  **int** y0 = getmaxy();    **float** dt = **0.1**;  **float** t = **0**;    **float** x=a\*t-b\*sin(t);  **float** y=a-b\*cos(t);  moveto(x\*xscale+x0,y0-y\*yscale);  setcolor(WHITE);  **while** (t<**30**) {  x=a\*t-b\*sin(t);  y=a-b\*cos(t);  lineto(x\*xscale+x0,y0-y\*yscale);  t+=dt;  }    getch();  **return** **0**;  } |

**float** r;

**float** scale = **1**; //scale to screen

**int** c;

printf("Enter r: ");

scanf("%f",&r);

printf("Enter color (1-15): ");

scanf("%i",&c);

//position of center

**int** x0 = getmaxx()/**2**;

**int** y0 = getmaxy()/**2**;

**float** x1 = r\*cos(pi/**4**);

**float** y1 = r\*sin(pi/**4**);

**float** x2 = r\*cos(pi+pi/**4**);

**float** y2 = r\*sin(pi+pi/**4**);

**int** xs1 = x0+x1\*scale;

**int** ys1 = y0+y1\*scale;

**int** xs2 = x0+x2\*scale;

**int** ys2 = y0+y2\*scale;

//draw ellipse

setcolor(c);

ellipse(x0,y0,**0**,**360**,r,r);

//draw rectangle

setcolor(WHITE);

rectangle(xs1,ys1,xs2,ys2);

getch();

**return** **0**;

}

Задание 5:

Разработать программу изображения точки, движущейся по косинусоиде

(y = cos(x)).

**Код программы:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | #include "graphics.h"  #include "stdio.h"  #include "math.h"  #include "time.h"  **const** **float** pi = **3.1415**;  **void** **draw**(**float** point) {  cleardevice();  **float** yscale = getmaxy()/**3**; //scale of y=cos(x)  **float** xscale = getmaxx()/**10**;    **int** x0 = **0**;  **int** y0 = getmaxy()/**2**;    **float** x = **0**;  **float** dx = **0.1**;  setcolor(**7**);  moveto(x\*xscale+x0,cos(x)\*yscale+y0);  setfillstyle(SOLID\_FILL, WHITE);  **while** (x<**10**) {  **if** (x==point) {  fillellipse(x\*xscale+x0,cos(x)\*yscale+y0,**10**,**10**);  }  **float** y = cos(x);  lineto(x\*xscale+x0,cos(x)\*yscale+y0);  x+=dx;  }    }  **int** **main**() {  **int** gd, gm;  gd = DETECT;  //graphics mode  initgraph(&gd, &gm, "");  **int** errorcode = graphresult();  //in case of error - print and exit  **if** (errorcode!=grOk) {  printf("Error: %s",grapherrormsg(errorcode));  **return** **1**;  }    **float** pointloc = **0**;  **while** (pointloc<**10**) {  draw(pointloc);  Sleep(**10**);  pointloc+=**0.1**;  }    getch();  **return** **0**;  } |

Задание 6:

Изобразить движущийся на зрителя экран (прямоугольник).

**Код программы:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75 | #include "graphics.h"  #include "stdio.h"  #include "math.h"  **const** **float** pi = **3.1415**;  **void** **rect**(**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, gm;  gd = DETECT;  //graphics mode  initgraph(&gd, &gm, "");  **int** errorcode = graphresult();  //in case of error - print and exit  **if** (errorcode!=grOk) {  printf("Error: %s",grapherrormsg(errorcode));  **return** **1**;  }    **int** sizex = **500**;  **int** sizey = **350**;  **float** factor = **0.3**;    //center  **int** x0 = getmaxx()/**2**;  **int** y0 = getmaxy()/**2**;    //edges  **int** x1 = x0-sizex/**2**;  **int** y1 = y0-sizey/**2**;  **int** x2 = x0+sizex/**2**;  **int** y2 = y0-sizey/**2**;  **int** x3 = x0+sizex/**2**;  **int** y3 = y0+sizey/**2**;  **int** x4 = x0-sizex/**2**;  **int** y4 = y0+sizey/**2**;    setcolor(WHITE);    **for** (**int** i=**0**;i<**20**;i++) {  rect(x1,y1,x2,y2,x3,y3,x4,y4);    //new point 1 between 1-2  **int** xn1 = x1+(x2-x1)\*factor;  **int** yn1 = y1+(y2-y1)\*factor;    //new point 2 between 2-3  **int** xn2 = x2+(x3-x2)\*factor;  **int** yn2 = y2+(y3-y2)\*factor;    //new point 3 between 3-4  **int** xn3 = x3+(x4-x3)\*factor;  **int** yn3 = y3+(y4-y3)\*factor;    //new point 4 between 4-1  **int** xn4 = x4+(x1-x4)\*factor;  **int** yn4 = y4+(y1-y4)\*factor;    x1=xn1; y1=yn1;  x2=xn2; y2=yn2;  x3=xn3; y3=yn3;  x4=xn4; y4=yn4;  }    getch();  **return** **0**;  } |