PROGRAM 1: DEVELOP CREATIVE NATURAL SCENERY OF SHIVA LINGA

INTRODUCTION

Natural scenery of Shiva linga is drawn besides the river which also includes some lotus in the river. Around the Shiva linga there are some trees and crows are flying in the sky and the sun is partially hidden behind the cloud.

Algorithm Adopted

- **Bezier Spline Curve Algorithm**: This Algorithm is used to create the sequence of flowers in various colors above the Shiva linga.
- **Ellipse Generation Algorithm:** This Algorithm is used to create snake in this scene and which is tan in color.
- **Circle Generation Algorithm:** This Algorithm is used to create a tree leaves which are green in color and cloud which are blue in color and sun is represented in orange color.

Description of Function

BUILT-IN FUNCTIONS:

- **GL_QUADS :** This function is used to create the river and used to create some of the parts in Shiva linga. It is used in stem of the tree.
- **GL_TRIANGLES :** This function is used to create stand of argarbatti and also in Shiva waterfall place.
- **GL_LINE_STRIP :** This function is used to create the birds in the sky.
- **GL_LINES:** This function is used to create the argarbatti lines. The smoke of argarbatti is also created by the lines.
- **glutBitmapString:** This function is used to create the text.
- **glRasterPos2i:** This function is used to set the position of the text.
- **glEnable(GL_LINE_STIPPLE):**This function is used to enable the line stipple which is used to generate the spacing the lines.

USER-DEFINED FUNCTIONS

- **River:** This function is used for creating the rectangle shape of an object which is filled with the blue color water.
- **Waves:** This function is used for creating the different types of line stipple. It increases the space between the lines. It will take an argument for different space and also takes starting and ending position of the lines.
- **Tree:** This function is used for creating the steam of the tree which is brown in color.
- **Outerandinnerleavesdisplay**: This function is used for creating the circle which also fills the circle completely by green color. The outer function will be used draw the circle from the edge of the inner circle
- **argarbatti:** This function is used for creating the triangle to support the lines which is drawn upon the triangles
- **light:** This function is used for creating deepa by drawing a circle. The circumference is divided by two. It consists of triangle to show the light and the flames are drawn using the lines and the line width is increased.
- **stand:** This function is used for creating the basement to the Shiva linga by using GL_QUADS which is black in color.
- **shivawaterfall:** Actually, In real scenario the fall will move from the Shiva body. In this project the water has not been included. But it the place where water will move from.
- **Shivamurthybody:** This function is used for creating the part of face of in Shiva linga using GL_QUADS which is black in color.
- **Shivamurthyface:** This function is used for creating the round face of the Shiva linga using the circle function which is black in color.
- **Mark and Kumkum:** This function is used for creating the lines and circle in the face of the Shiva linga. The lines are white in color and the circle is red in color.
- **Flower:** This function is used for creating it used to create the sequence of flowers using the Bezier Spline Curve Algorithm.
- **Lotus:** This function is used for creating the petals which are pink in color and the leaves which are green in color.
- **Birds:** This function is used for creating the birds using GL_LINE_STRIP which is black in color and which is present in the sky
- **Cloud**: This function is used for creating circles which will form the cloud which is blue in color.
- **Sun:** This function is used for creating the circle which will form the sun which is orange in color.

- **Snake:** This function is used for creating snake using the ellipse generating algorithm and the tongue of the snake is drawn with GL_LINE_STRIP which is red in color. Eyes of the snake are drawn in GL_POINTS which is black in color. The body of the snake is drawn using GL_QUADS which is tan in color.
- **Text:** This function is used for creating the text to display the contents of name, class, usn, title and college with different colors.
- **Delay:** This function is used for refer to first slide and refer to next slide which creates delay between the two slides.

.

SOURCE CODE

```
#include<GL/glut.h>
#include<stdio.h>
#include<math.h>


void init()
{
        glClearColor(0.0,0.0,0.0,0.0);
        glClear(GL_COLOR_BUFFER_BIT);
        gluOrtho2D(0,1200,0,600);
}



//................................................................RIVER........................................................................
void river()
{
        glBegin(GL_QUADS);
        {
                glColor3ub(32,178,170);
                glVertex2i(0,0); glVertex2i(1200,0); glVertex2i(1200,150);glVertex2i(0,150);
        glEnd();
        }
glFlush();
}
//.....................................................................................................................................................
```

```
//.............................................................WAVES............................................................
void waves()
{
	glColor3ub(0, 255, 255);

	glPushAttrib(GL_ENABLE_BIT);

	glLineStipple(35,0xAAAA);
	glEnable(GL_LINE_STIPPLE);
	 glBegin(GL_LINES);
		 glVertex2i(0,0); glVertex2i(1200,0);
	 glEnd();


	glLineStipple(30,0xAAAA);
	glEnable(GL_LINE_STIPPLE);
	glBegin(GL_LINES);
		glVertex2i(0,25); glVertex2i(1200,25);
	glEnd();


	 glLineStipple(25,0xAAAA);
	glEnable(GL_LINE_STIPPLE);
	glBegin(GL_LINES);
		 glVertex2i(0,50); glVertex2i(1200,50);
	glEnd();


	glLineStipple(20,0xAAAA);
	glEnable(GL_LINE_STIPPLE);
	glBegin(GL_LINES);
		glVertex2i(0,75); glVertex2i(1200,75);
	glEnd();

	glLineStipple(15,0xAAAA);
	glEnable(GL_LINE_STIPPLE);
	glBegin(GL_LINES);
		glVertex2i(0,100); glVertex2i(1200,100);
	glEnd();
```

```
        glLineStipple(10,0xAAAA);
        glEnable(GL_LINE_STIPPLE);
        glBegin(GL_LINES);
                glVertex2i(0,125); glVertex2i(1200,125);
        glEnd();

        glLineStipple(5,0xAAAA);
        glEnable(GL_LINE_STIPPLE);
        glBegin(GL_LINES);
                glVertex2i(0,140); glVertex2i(1200,140);
        glEnd();
glFlush();
glDisable(GL_LINE_STIPPLE);
}
//.........................................................................................................................


//..............................................................TREE STEAM....................................................

void tree()
{
        glBegin(GL_QUADS);
        {
                glColor3ub(128,0,0);
                glVertex2i(50,200); glVertex2i(65,200);
                glVertex2i(65,250);glVertex2i(50,250);
        glEnd();
        }




        glBegin(GL_QUADS);
        {
                glColor3ub(128,0,0);
                glVertex2i(100,150); glVertex2i(150,150);
                glVertex2i(150,400); glVertex2i(100,400);
        glEnd();
        }
```

```
          glBegin(GL_QUADS);
          {
                    glColor3ub(128,0,0);
                    glVertex2i(200,230); glVertex2i(210,230);
                     glVertex2i(210,260); glVertex2i(200,260);
          glEnd();
          }

          glBegin(GL_QUADS);
          {
                    glColor3ub(128,0,0);
                    glVertex2i(1005,200); glVertex2i(1015,200);
                    glVertex2i(1015,250); glVertex2i(1005,250);
          glEnd();
          }

          glBegin(GL_QUADS);
          {
                    glColor3ub(128,0,0);
                    glVertex2i(1100,150); glVertex2i(1150,150);
                     glVertex2i(1150,400); glVertex2i(1100,400);
          glEnd();
          }
glFlush();
}
//.........................................................................................................................
```

```
//..............................................................TREE LEAVES.........................................................
void innerleaves(float centerx, float centery, float r)
{
	int i, fill;
	float x, y;

	glColor3f(0.0,1.0,0.0);
	glPointSize(3.0);

	glBegin(GL_POINTS);
	for(fill=0; fill<=r; fill++)
	{
		for(i=0;i<360;i++)
		{
			x=fill*cosf(i); y=fill*sinf(i);
			glVertex2f(x+centerx, y+centery);
		}
	}
	glEnd();
glFlush();
}

void outerleaves(int centerx, int centery, int r)
{
	int i;
	float x, y;

	glColor3f(0.0,1.0,0.0);
	glPointSize(3.0);

	glBegin(GL_POINTS);
	for(i=0;i<360;i=i+20)
	{
		x=r*cosf(i); y=r*sinf(i);
		innerleaves(x+centerx, y+centery, 23);
	}
	glEnd();
}
```

```
void outerandinnerleavesdisplay()
{
        innerleaves(125, 400, 100);
        outerleaves(125, 400, 100);
        innerleaves(1125, 400, 100);
        outerleaves(1125, 400, 100);
        innerleaves(57, 250, 10);
        outerleaves(57, 250, 10);
        innerleaves(205, 260, 1);
        outerleaves(205, 260, 1);
        innerleaves(1010, 250, 10);
        outerleaves(1010, 250, 10);
}
//..........................................................................................................................



//.............................................................SHIVA ARGARBATTI.....,.................................
void argarbatti()
{
        glBegin(GL_TRIANGLES);
        {
                glColor3ub(105,105,105);
                glVertex2i(400,220); glVertex2i(440,220); glVertex2i(420,240);
        glEnd();
        }

        glLineWidth(3.0);
        glBegin(GL_LINES);
        {
                glColor3ub(0,0,0);
                glVertex2i(420,240); glVertex2i(400,280);
                glVertex2i(420,240); glVertex2i(440,280);
        glEnd();
        }
        glLineWidth(10.0);
        glBegin(GL_LINES);
        {
                glColor3ub(0,0,0);
                glVertex2i(400,285); glVertex2i(400,290);
                glVertex2i(400,295); glVertex2i(400,300);
                glVertex2i(440,285); glVertex2i(440,290);
                glVertex2i(440,295); glVertex2i(440,300);
        glEnd();
        }
glFlush();
}
//..............................................................................................................................



//.............................................................SHIVA DEEPA...................................................
void light(float centerx, float centery, float r)
{
        int i, fill;
        float x, y;

        glColor3ub(165,42,42);
        glPointSize(3.0);

        glBegin(GL_POINTS);
        for(fill=0;fill<=r;fill++)
        {
                for(i=180;i<360;i++)
                {
                     x=fill*cosf(i*3.142/180); y=fill*sinf(i*3.142/180);
                        glVertex2f(x+centerx, y+centery);
                }
        }
        glEnd();
glFlush();
}
```

```
void lightdisplay()
{
	light(330, 230, 15);
	light(370, 230, 15);
	light(750, 230, 15);
	light(790, 230, 15);
	light(830, 230, 15);
	light(870, 230, 15);
}

void fire()
{
	glBegin(GL_TRIANGLES);
	{
		glColor3ub(255,69,0);
		glVertex2i(320,230); glVertex2i(340,230); glVertex2i(330,240);
	glEnd();
	}

	glBegin(GL_TRIANGLES);
	{
		glColor3ub(255,69,0);
		glVertex2i(360,230); glVertex2i(380,230); glVertex2i(370,240);
	glEnd();
	}

	glBegin(GL_TRIANGLES);
	{
		glColor3ub(255,69,0);
		glVertex2i(740,230); glVertex2i(760,230); glVertex2i(750,240);
	glEnd();
	}

	glBegin(GL_TRIANGLES);
	{
		glColor3ub(255,69,0);
		glVertex2i(780,230); glVertex2i(800,230); glVertex2i(790,240);
	glEnd();
	}
```

```
        glBegin(GL_TRIANGLES);
        {
                glColor3ub(255,69,0);
                glVertex2i(820,230); glVertex2i(840,230); glVertex2i(830,240);
        glEnd();
        }

        glBegin(GL_TRIANGLES);
        {
                glColor3ub(255,69,0);
                glVertex2i(860,230); glVertex2i(880,230); glVertex2i(870,240);
        glEnd();
        }

        glLineWidth(10.0);
        glBegin(GL_LINES);
        {
                glColor3ub(255,69,0);
                glVertex2i(330,243); glVertex2i(330,260);
                glVertex2i(370,243); glVertex2i(370,260);
                glVertex2i(750,243); glVertex2i(750,260);
                glVertex2i(790,243); glVertex2i(790,260);
                glVertex2i(830,243); glVertex2i(830,260);
                glVertex2i(870,243); glVertex2i(870,260);
        glEnd();
        }
glFlush();
}
//...........................................................................................................................
```

```
//...............................................................SHIVA STAND..................................................
void stand()
{
            glBegin(GL_QUADS);
            {
                        glColor3f(0.0,0.0,0.0);
                        glVertex2i(200,150); glVertex2i(1000,150);
                         glVertex2i(900,220); glVertex2i(300,220);
            glEnd();
            }

            glBegin(GL_QUADS);
            {
                        glColor3f(0.0,0.0,0.0);
                        glVertex2i(550,300); glVertex2i(650,300);
                         glVertex2i(700,350); glVertex2i(500,350);
            glEnd();
            }

            glBegin(GL_QUADS);
            {
                        glColor3f(0.0,0.0,0.0);
                        glVertex2i(500,250); glVertex2i(700,250);
                        glVertex2i(650,300); glVertex2i(550,300);
            glEnd();
            }

            glBegin(GL_QUADS);
            {
                        glColor3f(0.0,0.0,0.0);
                        glVertex2i(500,220); glVertex2i(700,220);
                         glVertex2i(700,250); glVertex2i(500,250);
            glEnd();
            }
glFlush();
}
//..............................................................................................................................................
```

```
//..........................................................SHIVA WATER FALL.......................................
void shivawaterfall()
{
        glBegin(GL_QUADS);
        {
                glColor3f(0.0,0.0,0.0);
                glVertex2i(450,350); glVertex2i(750,350);
                 glVertex2i(750,400); glVertex2i(450,400);
        glEnd();
        }

        glBegin(GL_QUADS);
        {
                glColor3f(0.0,0.0,0.0);
                glVertex2i(715,340); glVertex2i(760,340);
                 glVertex2i(760,400); glVertex2i(715,400);
        glEnd();
        }

        glBegin(GL_TRIANGLES);
        {
                glColor3f(0.0,0.0,0.0);
                glVertex2i(715,340); glVertex2i(760,340); glVertex2i(740,310);
        glEnd();
        }
glFlush();
}
//.............................................................................................................................
```

```
//...........................................................SHIVA FACE AND BODY....................................
void shivamurthybody()
{
        glBegin(GL_QUADS);
        {
                glColor3f(0,0,0);
                glVertex2i(560,350); glVertex2i(640,350);
                glVertex2i(640,420); glVertex2i(560,420);
        glEnd();
        }
glFlush();
}


void shivamurthyface(float centerx, float centery, float r)
{
        int i, fill;
        float x, y;

        glColor3f(0,0,0);
        glPointSize(3.0);

        glBegin(GL_POINTS);
        for(fill=0;fill<=r;fill++)
        {
                for(i=0;i<360;i++)
                {
                        x=fill*cosf(i); y=fill*sinf(i);
                        glVertex2f(x+centerx, y+centery);
                }
        }
        glEnd();
glFlush();
}
void shivamurthyfacedisplay()
{
        shivamurthyface(600, 425, 40);
}
//..........................................................................................................................
```

```
//...............................................................SHIVA KUMKUM............................................
void mark()
{
	glLineWidth(5.0);
	glBegin(GL_LINES);
	{
		glColor3f(1.0,1.0,1.0);
		glVertex2i(570,400); glVertex2i(620,400);
		glVertex2i(570,420); glVertex2i(620,420);
		glVertex2i(570,440); glVertex2i(620,440);
	glEnd();
	}
glFlush();
}

void kumkum(float centerx, float centery, float r)
{
	int i, fill;
	float x, y;
	glColor3ub(255,0,0);
	glPointSize(3.0);
	glBegin(GL_POINTS);
	for(fill=0;fill<=r;fill++)
	{
		for(i=0;i<360;i++)
		{
			x=fill*cosf(i); y=fill*sinf(i);
			glVertex2f(x+centerx, y+centery);
		}
	}
	glEnd();
glFlush();
}

void kumkumdisplay()
{
	kumkum(595, 420, 10);
}
//...........................................................................................................................
```

```
//............................................................SHIVA'S FLOWERS.............................................
void flower(int x[],int y[], int limit, int a)
{
        int i;

        glLineWidth(2.0);

        for(i=0;i<limit;i+=5)
        {
                glBegin(GL_LINES);
                glVertex2i(x[i]+a,y[i]); glVertex2i(x[i]-a,y[i]);
                glVertex2i(x[i],y[i]+a); glVertex2i(x[i],y[i]-a);
                glVertex2i(x[i]+a,y[i]+a); glVertex2i(x[i]-a,y[i]-a);
                glVertex2i(x[i]+a,y[i]-a); glVertex2i(x[i]-a,y[i]+a);
        glEnd();
        }
glFlush();
}

void leaves(int x0,int y0,int x1,int y1,int x2,int y2,int x3,int y3)
{
        int ax,ay,bx,by,cx,cy,dx,dy;
        int x[100],y[100],i,k;
        double step,t;
        float size;

        ax=-x0+3*x1-3*x2+x3;
        ay=-y0+3*y1-3*y2+y3;

        bx=3*x0-6*x1+3*x2;
        by=3*y0-6*y1+3*y2;

        cx=-3*x0+3*x1;
        cy=-3*y0+3*y1;

        dx=x0;
        dy=y0;

        step=100.0;
        size=1.0/(double)step;
        for(i=1;i<step;i++)
        {
                t=size*(double)i;
                x[i]=(ax*t*t*t)+(bx*t*t)+(cx*t)+dx;
                y[i]=(ay*t*t*t)+(by*t*t)+(cy*t)+dy;
        }
        flower(x,y,i,5);
}


void leavesdisplay()
{
        glColor3ub(255,255,0);
        leaves(560,350, 525,510, 665,525, 640,350);

        glColor3ub(255,0,0);
        leaves(550,350, 515,510, 675,525, 650,350);

        glColor3ub(255,255,0);
        leaves(540,350, 505,510, 685,525, 660,350);

        glColor3ub(255,0,0);
        leaves(530,350, 495,510, 695,525, 670,350);

}
//.......................................................................................................................
```

```
//...............................................................LOTUS.....................................................................
void lotus(int x, int y)
{
        glBegin(GL_TRIANGLES);
        {
                glColor3ub(0,128,0);
                glVertex2i(x,y-45); glVertex2i(x-15,y); glVertex2i(x+15,y);

                glColor3ub(0,255,0);
                glVertex2i(x,y-15); glVertex2i(x,y+15); glVertex2i(x-45,y);
                glVertex2i(x,y-15); glVertex2i(x+45,y); glVertex2i(x,y+15);

                glColor3ub(255,0,255);
                glVertex2i(x-15,y); glVertex2i(x+15,y); glVertex2i(x,y+55);
                glVertex2i(x-20,y); glVertex2i(x,y); glVertex2i(x-35,y+45);
                glVertex2i(x+20,y); glVertex2i(x+10,y); glVertex2i(x+35,y+45);
                glVertex2i(x,y); glVertex2i(x-20,y+35); glVertex2i(x-20,y);
                glVertex2i(x,y); glVertex2i(x+20,y); glVertex2i(x+20,y+45);
        glEnd();
        }
glFlush();
}

void lotusdisplay()
{
        lotus(100,10);
        lotus(100,100);
        lotus(200,50);
        lotus(400,120);
        lotus(500,70);
        lotus(800,50);
        lotus(1000,80);
}
//.........................................................................................................................................
```

```
//..................................................................BIRDS...................................................................
void birds()
{
        glLineWidth(5.0);
        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(290,500); glVertex2i(300,490); glVertex2i(310,500);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(320,480); glVertex2i(330,470); glVertex2i(340,480);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(350,460); glVertex2i(360,450); glVertex2i(370,460);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(380,480); glVertex2i(390,470); glVertex2i(400,480);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(410,500); glVertex2i(420,490); glVertex2i(430,500);
        glEnd();
        }
        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(100,500); glVertex2i(110,490); glVertex2i(120,500);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(150,450); glVertex2i(160,440); glVertex2i(170,450);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(900,450);glVertex2i(910,440);glVertex2i(920,450);
        glEnd();
        }

        glBegin(GL_LINE_STRIP);
        {
                glColor3ub(0,0,0);
                glVertex2i(1000,550); glVertex2i(1010,540); glVertex2i(1020,550);
        glEnd();
        }
glFlush();
}
//......................................................................................................................
```

```
//....................................................................CLOUD....................................................................
void cloud(float centerx, float centery, float r)
{
        int i, fill;
        float x, y;
        glColor3ub(0,206,209);
        glPointSize(3.0);

        glBegin(GL_POINTS);
        for(fill=0;fill<=r;fill++)
        {
                for(i=0;i<360;i++)
                {
                      x=fill*cosf(i); y=fill*sinf(i);
                         glVertex2f(x+centerx, y+centery);
                }
        }
        glEnd();
glFlush();
}

void cloudisplay()
{
        cloud(40, 580, 80);
        cloud(300, 550, 20);
        cloud(310, 570, 20);
        cloud(320, 550, 20);
        cloud(600, 580, 25);
        cloud(610, 550, 25);
        cloud(620, 570, 25);
        cloud(800, 500, 20);
        cloud(810, 520, 20);
        cloud(820, 500, 10);
        cloud(810, 480, 20);
        cloud(1000, 600, 30);
        cloud(1010, 620, 50);
        cloud(1020, 600, 20);
        cloud(1010, 580, 20);
}
//......................................................................................................................................
```

```
//..............................................................SUN.........................................................
void sun(float centerx, float centery, float r)
{
        int i, fill;
        float x, y;

        glColor3ub(255,69,0);
        glPointSize(3.0);

        glBegin(GL_POINTS);
        for(fill=0;fill<=r;fill++)
        {
                for(i=0;i<360;i++)
                {
                      x=fill*cosf(i); y=fill*sinf(i);
                         glVertex2f(x+centerx, y+centery);
                }
        }
        glEnd();
glFlush();
}

void sundisplay()
{
        sun(350, 550, 40);
}
//.........................................................................................................................
```

```
//....................................................................SNAKE....................................................................
void snakepixel(int x, int y)
{
        glColor3ub(210,180,140);
        glPointSize(2.0);

        glBegin(GL_POINTS);
                glVertex2i(x,y);
        glEnd();
glFlush();
}

void snakeface(int xc, int yc,int xr, int yr)
{
        int rx=xr/2, ry=yr/2;
        int fill;
        float theta;

        for(fill=0;fill<rx--,ry--;fill++)
        {
                for(theta=0;theta<=180;theta+=0.5)
                {
                        int x=xc+rx*cos(theta); int y=yc+ry*sin(theta);
                        snakepixel(x,y);
                }
        }
}

void snaketail()
{
        int x=600, y=550;
        glColor3ub(210,180,140);
        glBegin(GL_QUADS);
        {
                glVertex2i(x-15,y-45);
                glVertex2i(x+15,y-45);
                glVertex2i(x+5,y-100);
                glVertex2i(x-5,y-100);
        glEnd();
        }
        glColor3ub(255,222,173);
        {
                glBegin(GL_QUADS);
                glVertex2i(x-5,y-2); glVertex2i(x+5,y-2);
                glVertex2i(x+2,y-100); glVertex2i(x-2,y-100);
        glEnd();
        }
glFlush();
}

void snakeeye()
{
        glPointSize(5.0);
        glBegin(GL_POINTS);
        {
                glVertex2i(588,525); glVertex2i(612,525);
        }
        glEnd();
        glFlush();
}

void snaketongue()
{
        glColor3f(1.0, 0.0, 0.0);
        glLineWidth(5.0);
        glBegin(GL_LINE_STRIP);
        {
                glVertex2i(595,540); glVertex2i(600,550); glVertex2i(605,540);
        }
        glEnd();
        glFlush();
}
void snakedisplay()
{
        snakeface(600, 525, 55, 50); snakeface(600, 550, 22, 20);
        snakeeye();
        snaketail();
        snaketongue();
}
//..........................................................................................................................
```

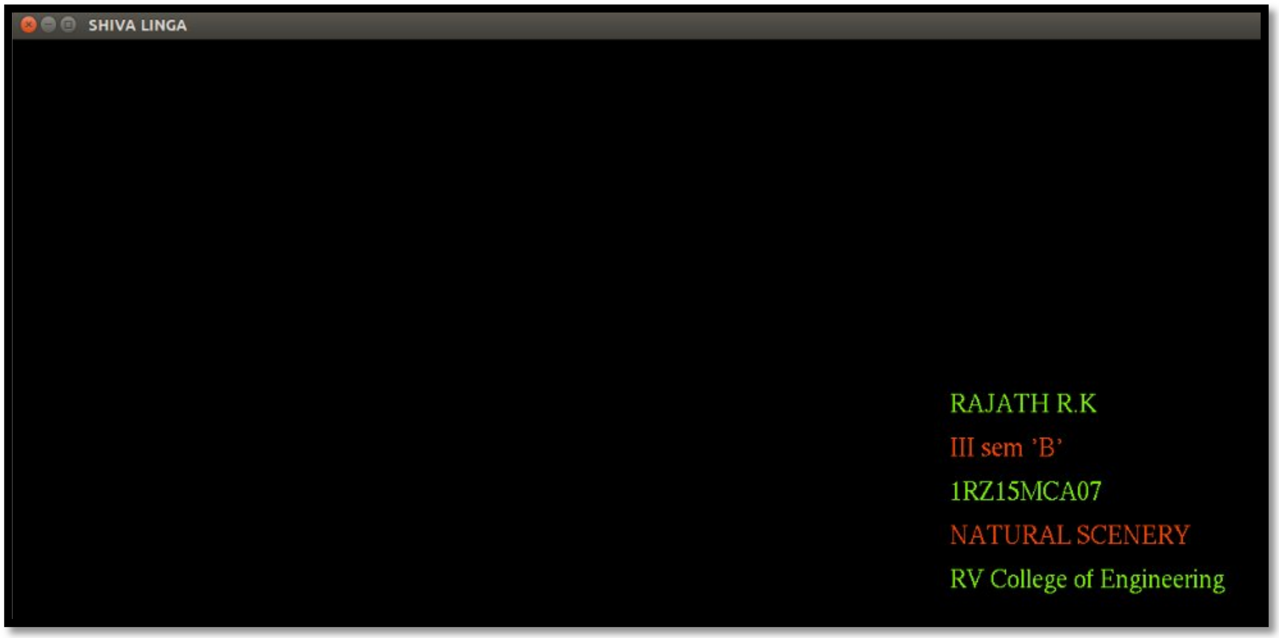
```
//...............................................................TEXT.........................................................................
void text()
{
	char name[50]      ="RAJATH R.K";
	char class[50]     ="III sem 'B'";
	char usn[50]       ="1RZ15MCA07";
	char title[50]     ="NATURAL SCENERY";
	char college[50]   ="RV College of Engineering";

	glColor3ub(124,252,0);
	glRasterPos2i(900,260);
	glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char
*)name);

	glColor3ub(255,69,0);
	glRasterPos2i(900,220);
	glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char *)class);

	glColor3ub(124,252,0);
	glRasterPos2i(900,180);
	glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char *)usn);

	glColor3ub(255,69,0);
	glRasterPos2i(900,140);
	glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char *)title);



	glColor3ub(124,252,0);
	glRasterPos2i(900,100);
	glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char
*)college);
glFlush();
}

void delay()
{
	int i;
	for(i=0;i<1000000000;i++) {}
}
```

```
void displaytext()
{
	text();
}

void delaytext()
{
	glClearColor(1.0,1.0,1.0,1.0);
	displaytext();
	delay(); delay();
	glClear(GL_COLOR_BUFFER_BIT);
}
//..................................................................................................................................
```

```
//.............................................................DISPLAY MAIN.........................................................
void display()
{
        delaytext();

        sundisplay();
        cloudisplay();

        tree();
        outerandinnerleavesdisplay();

        argarbatti();
        lightdisplay();
        fire();

        snakedisplay();

        stand();
        shivawaterfall();
        shivamurthybody();
        shivamurthyfacedisplay();
        mark();
        kumkumdisplay();
        leavesdisplay();

        birds();

        river();
        waves();

        lotusdisplay();
}
//.................................................................................................................................
```

```
//...................................................................MAIN PROGRAM...........................................
int main(int argc,char **argv)
{

                glutInit(&argc,argv);
                glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
                glutInitWindowSize(1200,600);
                glutInitWindowPosition(10,10);
                glutCreateWindow("SHIVA LINGA");

                init();
                glutDisplayFunc(display);
                glutMainLoop();

        return 0;
}
//......................................................................................................................................
```

OUTPUT



SHIVA LINGA

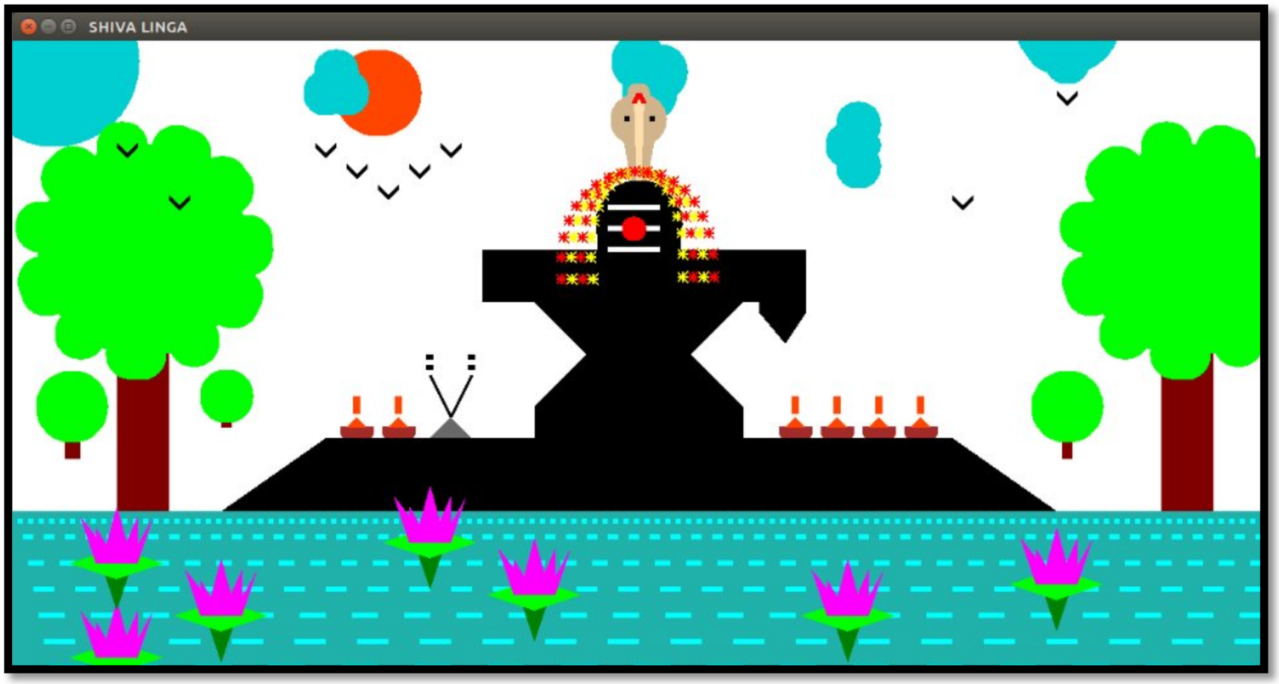
RAJATH R.K

III sem 'B'

1RZ15MCA07

NATURAL SCENERY

RV College of Engineering



PROGRAM 2 : MID-POINT CIRCLE GENERATION ALGORITHM FOR OLYMPIC SYMBOLS

INTRODUCTION

Different types of Olympic circles are drawn in this program. It consists of text which displays the heading of the program. The circles are drawn in different colors with different interval of time.

Algorithm Adopted

- **Circle Generation Algorithm:** This Algorithm is used to create different types of circle which is in different colors and also which is in different shapes. i.e., set x, y position and radius.

Description of Function

BUILT-IN FUNCTIONS

- **GL_LINES:** This function is used to create the lines under the text from one end to anthor end.
- **glutBitmapString:** This function is used to create the text.
- **glRasterPos2i:** This function is used to set the position of the text.

USER-DEFINED FUNCTIONS

- **Text:** This function is used to create the text in the program.The line is used to underline the text.
- **Circle:** This function is used to create the circle in five different colors. The position and radius of the circle will be specified by the user.
- **Delay:** This function is used to display the objects in different interval of time.

SOURCE CODE

```
#include<GL/glut.h>
#include<stdio.h>
#include<math.h>

void init()
{
        glClearColor(1.0,1.0,1.0,1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        gluOrtho2D(0,1200,0,600);
}

//...............................................................TEXT.........................................................................
void text()
{
        char name[20]="OLYMPIC SYMBOLS";
        glColor3ub(128,0,0);

        glRasterPos2i(550,500);
        glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (const unsigned char
*)name);
}

void underline()
{
        glLineWidth(3.0);
        glBegin(GL_LINES);
        {
                glVertex2i(530,490); glVertex2i(790,490);
        glEnd();
        }
glFlush();
}

void displaytext()
{
        text(); underline();
}
//.........................................................................................................................................
```

```
//............................................................CIRCLE....................................................................
void setpixel(int x,int y)
{
        glPointSize(2.0);
        glBegin(GL_POINTS);
                glVertex2i(x,y);
        glEnd();
glFlush();
}

void putpixel(int xc,int yc,int x,int y)
{
        setpixel(xc+x,yc+y); setpixel(xc+y,yc+x);
        setpixel(xc-x,yc+y); setpixel(xc-y,yc+x);
        setpixel(xc-x,yc-y); setpixel(xc-y,yc-x);
        setpixel(xc+x,yc-y); setpixel(xc+y,yc-x);
}

void circle(int xc,int yc,int r)
{
        int p=1-r,x=0,y=r;
        putpixel(xc,yc,x,y);

        while(x<y)
        {
                x++;
                if(p<0)
                {
                        p=p+2*x+1;
                }
                else
                {
                        y--;
                        p=p+2*(x-y)+1;
                }
                putpixel(xc,yc,x,y);
        }
glFlush();
}
```

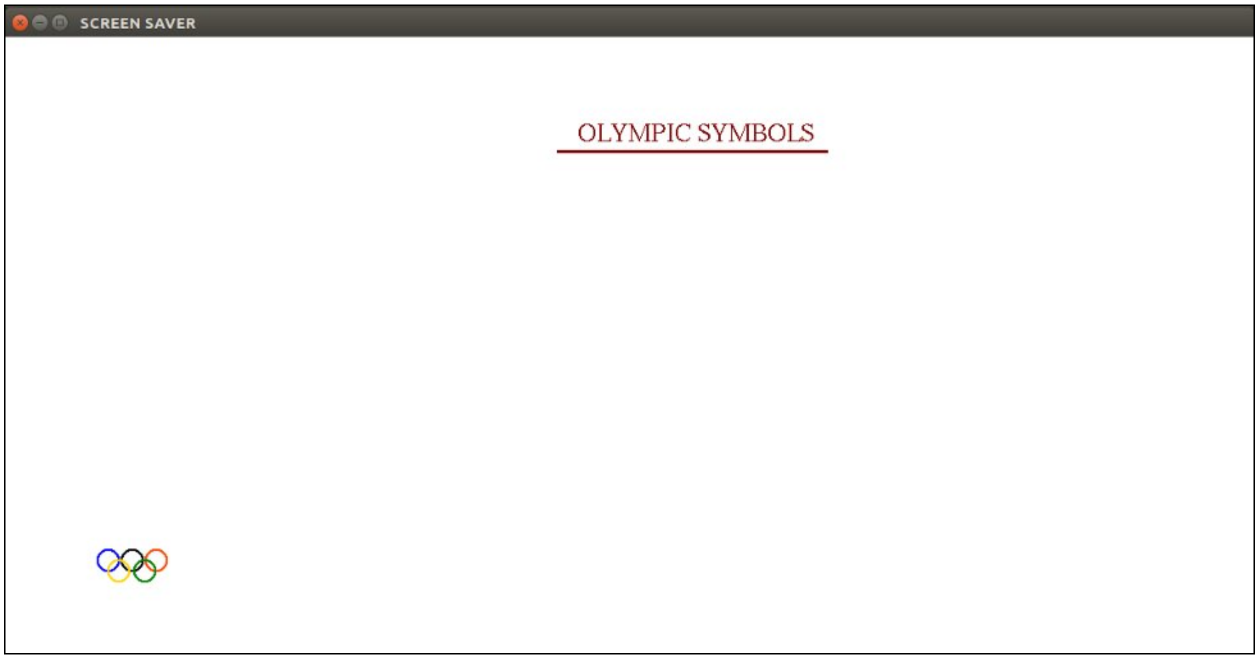
```
void displaycircle1()
{
	glColor3ub(0,0,255);
				circle(100,100,10);
	glColor3ub(0,0,0);
				circle(123,100,10);
	glColor3ub(255,69,0);
				circle(146,100,10);
	glColor3ub(255,215,0);
				circle(110,90,10);
	glColor3ub(0,128,0);
				circle(135,90,10);
}

void displaycircle2()
{

	glColor3ub(0,0,255);
				circle(1000,100,20);
	glColor3ub(0,0,0);
				circle(1045,100,20);
	glColor3ub(255,69,0);
				circle(1090,100,20);
	glColor3ub(255,215,0);
				circle(1020,80,20);
	glColor3ub(0,128,0);
				circle(1070,80,20);
}

void displaycircle3()
{
	glColor3ub(0,0,255);
				circle(100,450,10);
	glColor3ub(0,0,0);
				circle(123,450,10);
	glColor3ub(255,69,0);
				circle(146,450,10);
	glColor3ub(255,215,0);
				circle(110,440,10);
```

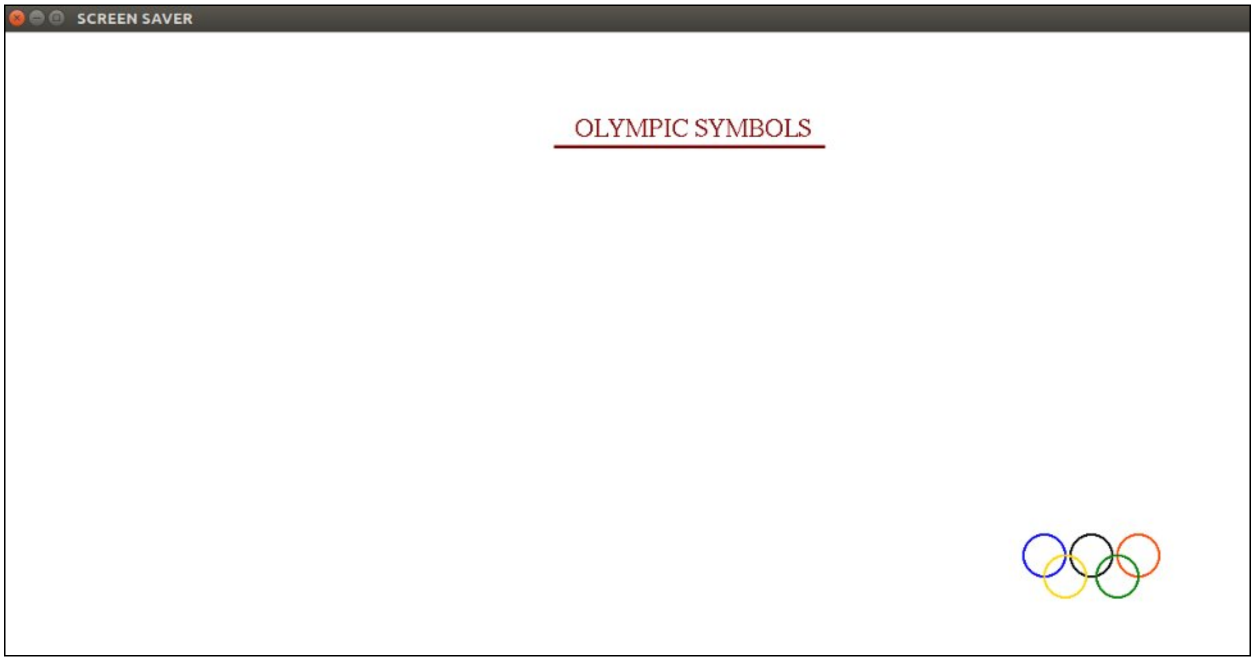
```
        glColor3ub(0,128,0);
                                circle(135,440,10);
}

void displaycircle4()
{
        glColor3ub(0,0,255);
                                circle(1000,450,20);

        glColor3ub(0,0,0);
                                circle(1045,450,20);
        glColor3ub(255,69,0);
                                circle(1090,450,20);
        glColor3ub(255,215,0);
                                circle(1020,430,20);
        glColor3ub(0,128,0);
                                circle(1070,430,20);
}

void displaycircle5()
{
        glColor3ub(0,0,255);
                                circle(500,275,50);
        glColor3ub(0,0,0);
                                circle(610,275,50);
        glColor3ub(255,69,0);
                                circle(720,275,50);
        glColor3ub(255,215,0);
                                circle(545,230,50);
        glColor3ub(0,128,0);
                                circle(665,230,50);
}

void delay()
{
        int i;
        for(i=0;i<1000000000;i++) {}
}
```

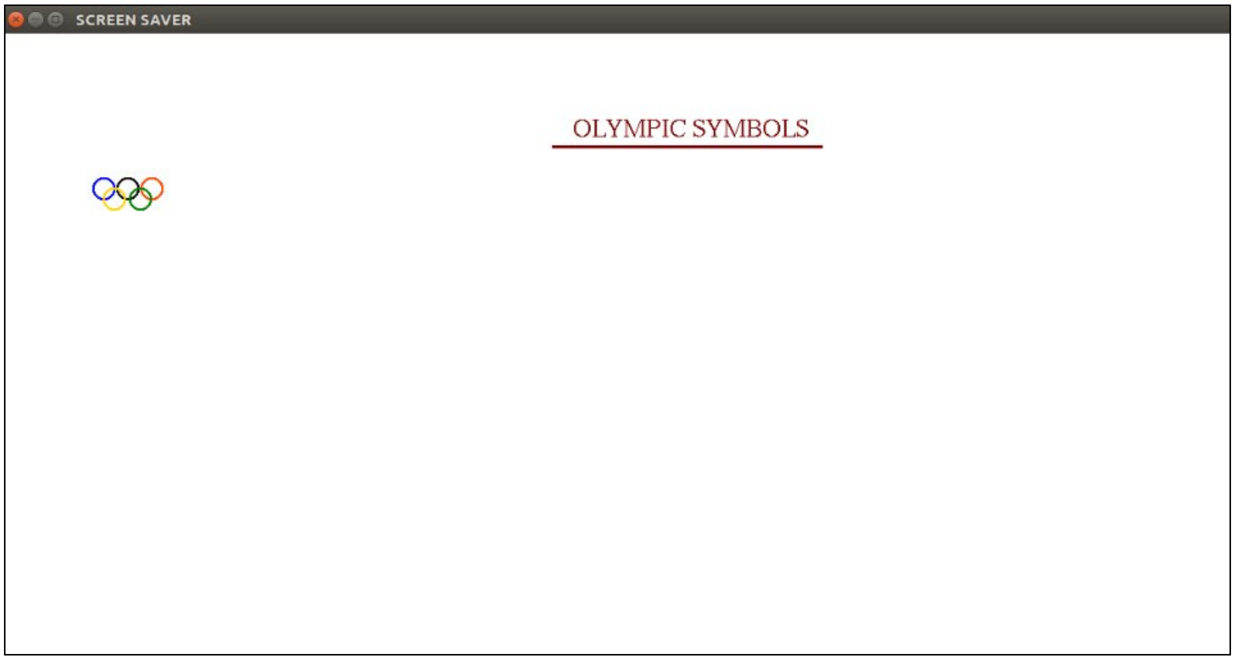
```
void circles()
{
        displaytext();

        displaycircle1();
                delay();
                glClear(GL_COLOR_BUFFER_BIT);
                displaytext();

        displaycircle2();
                delay();
                glClear(GL_COLOR_BUFFER_BIT);
                displaytext();

        displaycircle3();
                delay();
                glClear(GL_COLOR_BUFFER_BIT);
                displaytext();

        displaycircle4();
                delay();
                glClear(GL_COLOR_BUFFER_BIT);
                displaytext();

        displaycircle5();
                delay();
                displaytext();
}
//......................................................................................................................


//..........................................................DISPLAY MAIN....................................................
void display()
{
        circles();
}
//......................................................................................................................
```

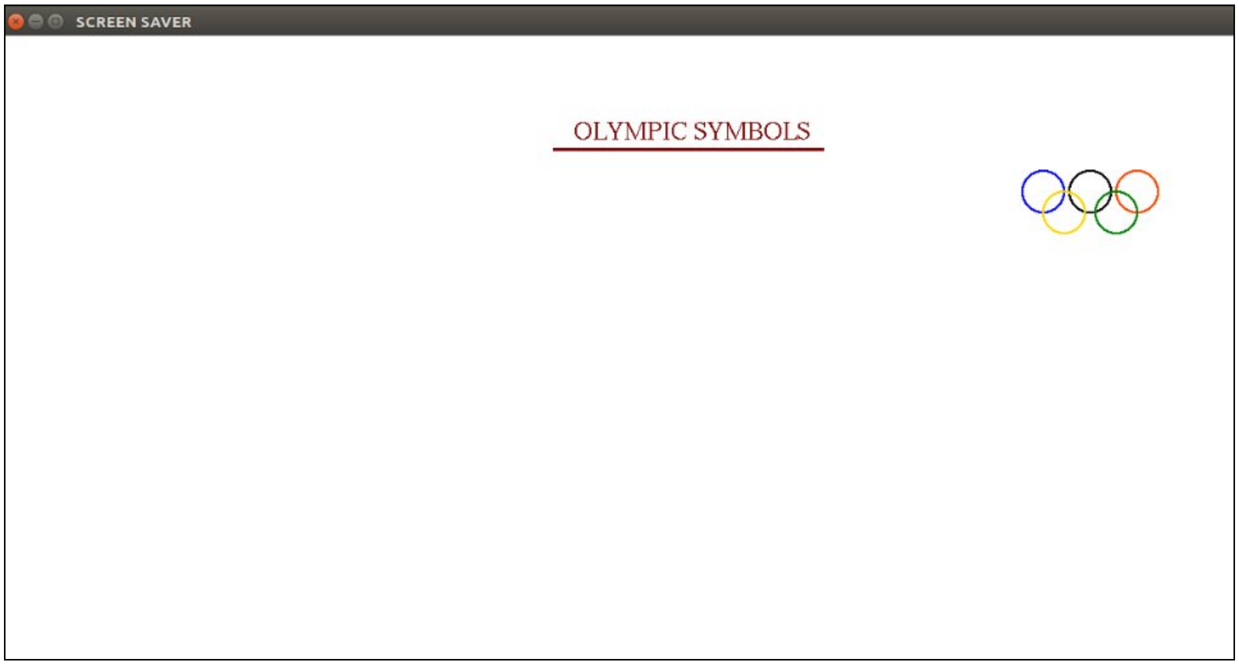
```
//.................................................................MAIN PROGRAM...................................................
int main(int argc,char **argv)
{

			glutInit(&argc,argv);
			glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
			glutInitWindowSize(1200,600);
			glutInitWindowPosition(10,10);
			glutCreateWindow("SCREEN SAVER");

			init();
			glutDisplayFunc(display);
			glutMainLoop();

	return 0;
}
//......................................................................................................................................
```

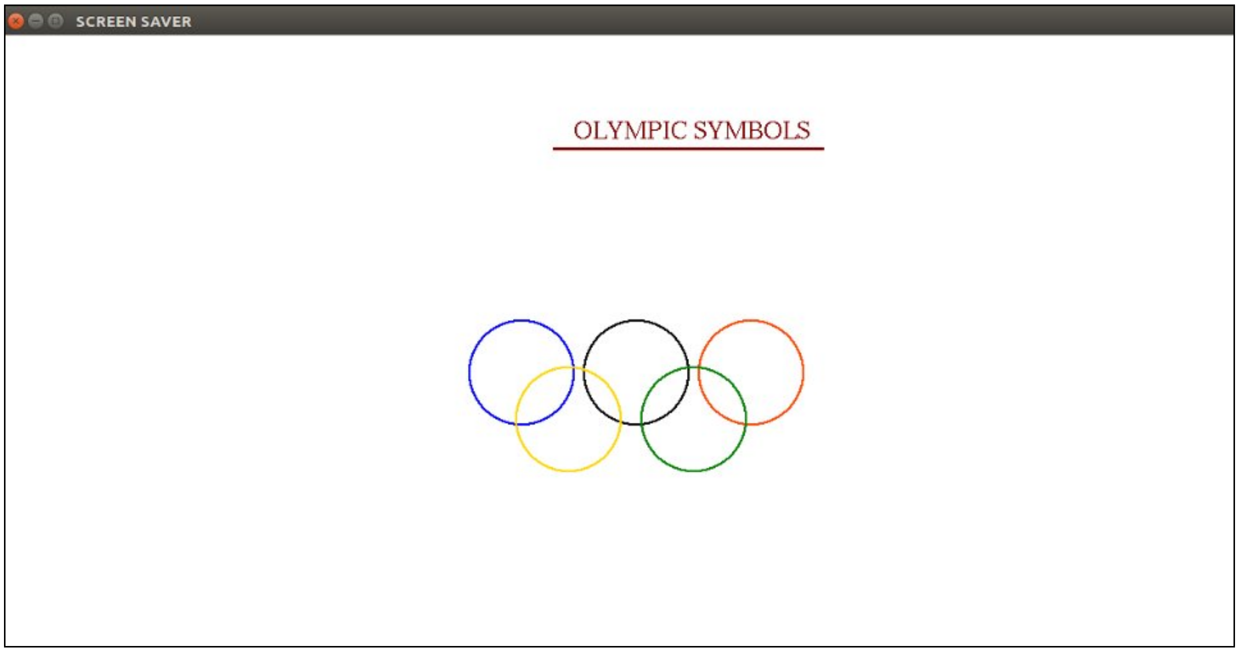
OUPUT:


SCREEN SAVER
OLYMPIC SYMBOLS


SCREEN SAVER
OLYMPIC SYMBOLS


SCREEN SAVER
OLYMPIC SYMBOLS


SCREEN SAVER
OLYMPIC SYMBOLS


SCREEN SAVER
OLYMPIC SYMBOLS