PROGRAM -8

8. Develop a menu driven program to animate a flag using Bezier Curve algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#include<GL/glut.h>
#include<math.h>
#include<iostream>
# define PI 3.1416
GLsizei w=600,h=600;
GLfloat xmin=0.0,xmax=120.0;
GLfloat ymin=0.0,ymax=120.0;
#define wave 1
#define stop 2
#define quit 3
class w3d {
public:
GLfloat x,y,z;
};

void bino(GLint n,GLint* c)
{
GLint k,j;
for(k=0;k<=n;k++)
{
c[k]=1;
for(j=n;j>=k+1;j--)
c[k]*=j;
for(j=n-k;j>=2;j--)
c[k]/=j;
}
}

void compute(GLfloat u,w3d *bp,GLint ncp,w3d *cp,GLint *c)
{
GLint k,n=ncp-1;
GLfloat bbf;
bp->x=bp->y=bp->z=0.00;
for(k=0;k<ncp;k++)
{
```

```
bbf=c[k]*pow(u,k)*pow(1-u,n-k);
bp->x+=cp[k].x*bbf;
bp->y+=cp[k].y*bbf;
bp->z+=cp[k].z*bbf;
}
}

void bezier(w3d *cp,GLint ncp,GLint nbcp)
{
w3d bcp;
GLfloat u;
GLint *c,k;
c=new GLint[ncp];
bino(ncp-1,c);
glBegin(GL_LINE_STRIP);
for(k=0;k<=nbcp;k++)
{
u=(GLfloat)k/(GLfloat)nbcp;
compute(u,&bcp,ncp,cp,c);
glVertex2f(bcp.x,bcp.y);
}
glEnd();
delete [ ] c;
}
static float t=0;
void display()
{
GLint i, ncp=4,nbcp=20;

w3d cp[4]={
{20,100,0},
{30,110,0},
{50,90,0},
 {60,100,0}};

 cp[1].x+=10*sin(t*PI/180.0);
 cp[1].y+=5*sin(t*PI/180.0);
 cp[2].x-=10*sin((t+30)*PI/180.0);
 cp[2].y-=10*sin((t+30)*PI/180.0);
 cp[3].x-=4*sin((t)*PI/180.0);
```

```
cp[3].y+=sin((t-30)*PI/180.0);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,1.0,1.0);
glPointSize(5);
glPushMatrix();
glLineWidth(5);
glColor3f(255/255,153/255.0,51/255.0);
for(i=0;i<8;i++)
{
glTranslatef(0,-0.8,0);
bezier(cp,ncp,nbcp);
}
glColor3f(1,1,1);
for(i=0;i<8;i++)
{
glTranslatef(0,-0.8,0);
bezier(cp,ncp,nbcp);
}
glColor3f(19/255.0,136/255.0,8/255.0);
for(i=0;i<8;i++)
{
glTranslatef(0,-0.8,0);
bezier(cp,ncp,nbcp);
}
glPopMatrix();
glColor3f(0.7,0.5,0.3);
glLineWidth(5);
glBegin(GL_LINES);
glVertex2f(20,100);
glVertex2f(20,40);
glEnd();
glFlush();
glutPostRedisplay();
glutSwapBuffers();
}

void res(GLint nW,GLint nH)
{
glViewport(0,0,nW,nH);
glMatrixMode(GL_PROJECTION);
```

```
glLoadIdentity();
gluOrtho2D(xmin,xmax,ymin,ymax);
glMatrixMode(GL_MODELVIEW);
glutPostRedisplay();
}

void animate()
{
t+=5;
glutPostRedisplay();
}

void menu(int item)
{
switch(item)
{
case wave:
glutIdleFunc(animate);
break;
case stop:
glutIdleFunc(NULL);
break;
case quit:
exit(0);
break;
}
}

int main(int argc,char** argv)
{
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
glutInitWindowPosition(50,50);
glutInitWindowSize(w,h);
glutCreateWindow("flag animation");
glutDisplayFunc(display);
glutReshapeFunc(res);
glutCreateMenu(menu);
glutAddMenuEntry("flag waiving",wave);
glutAddMenuEntry("stop waiving",stop);
glutAddMenuEntry("quit",quit);
```

```
glutAttachMenu(GLUT_RIGHT_BUTTON);
glutMainLoop();
}
```

Output command

To create file - gedit filename.c

To compile file - gcc filename.c -lGL -lGLU -lglut

To execute - ./a.out