```
//CG Lab 06 Open GL Queue



#include<iostream>
#include<Windows.h>
#include<GL/glut.h>
#include<string>
#include<sstream>
using namespace std;

int capacity = 10;

int que[10] = {};
int data;
  static int Qsize = 0;



void display()
{
    int x=50;
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(10.0);
    glColor3f(0.0,0.0,0.0);

    for(int j = 0; j < Qsize; j++){

    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    glBegin(GL_POLYGON);
    glVertex2f(x,100);
    glVertex2f(x+50,100);
    glVertex2f(x+50,50);
    glVertex2f(x,50);
    glEnd();

    glRasterPos2f(x+10, 70);

    int val = que[j];
    ostringstream stream;
    stream<<val;
    string s = stream.str();
    for(int i= 0; i<3;i++){
    glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, s[i]);
    }
    x+=70;
    }
    glutSwapBuffers();
        glPopMatrix();
    glEnd();
    glFlush();
}

void enQueue(){
if(Qsize<10){
```

```
cout<<"Enter number to be added in queue: ";
cin>>data;
que[Qsize] = data;
Qsize += 1;
}else{
cout<<"Limit of queue has been exceed!";
}
}

void deQueue(){
if(Qsize > 0){

    for(int i=0;i<=Qsize-1;i++)
    {
            que[i]=que[i+1];
    }
    que[capacity]={};
    cout<<"Element deleted successfully!"<<endl;
    Qsize -= 1;
}else{
cout<<"No element present in queue!"<<endl;
}
}

void keyboard(unsigned char key, int x, int y)
{
        switch (key)
        {
        case 'e' | 'E': enQueue(); display();
               break;

        case 'd' | 'D': deQueue(); display();
               break;
        case 'x' | 'X': exit(0);
        }
}

void Qfront(){
if(Qsize > 0){
    cout<<"The front element is: "<<que[0]<<endl;
}else{
cout<<"No element present in queue!"<<endl;
}
}
void Qrear(){
if(Qsize > 0){
    cout<<"The last element is: "<<que[Qsize - 1]<<endl;
}else{
cout<<"No element present in queue!"<<endl;
}
}


void init()
{
    glClearColor(1,1,1,1);
}
```

```
void reshape(int w, int h)
{
    glViewport(0,0,w,h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0,500,0,500);
    glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char*argv[])
{

    int option = 1;

    //Queue element;
    while(7>option>0){
    cout<<"\n\n<---------- MENU ---------->"<<endl;
    cout<<"Enter operation to be performed on queues: "<<endl;
    cout<<"1. enQueue \n2. deQueue \n3. Front \n4. Rear \n5. Size \n6.
Display"<<endl;
    cout<<"Enter option (0 to exit): ";
    cin>>option;
    switch(option){
    case 1: enQueue();
            break;
    case 2: deQueue();
            break;
    case 3: Qfront();
            break;
    case 4: Qrear();
            break;
    case 5: cout<<"Total element/s in queue is/are: "<<Qsize;
            break;
    case 6: glutInit(&argc, argv);
            glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
            glutInitWindowPosition(0,0);
            glutInitWindowSize(500,500);
            glutCreateWindow("Queues Using OpenGL");
            glutDisplayFunc(display);
            glutKeyboardFunc(keyboard);
            glutReshapeFunc(reshape);
            init();
            glutMainLoop();
            break;
    default: cout<<"Program ended successfully!"<<endl;
            break;
    }
    }
    return 0;
}
```