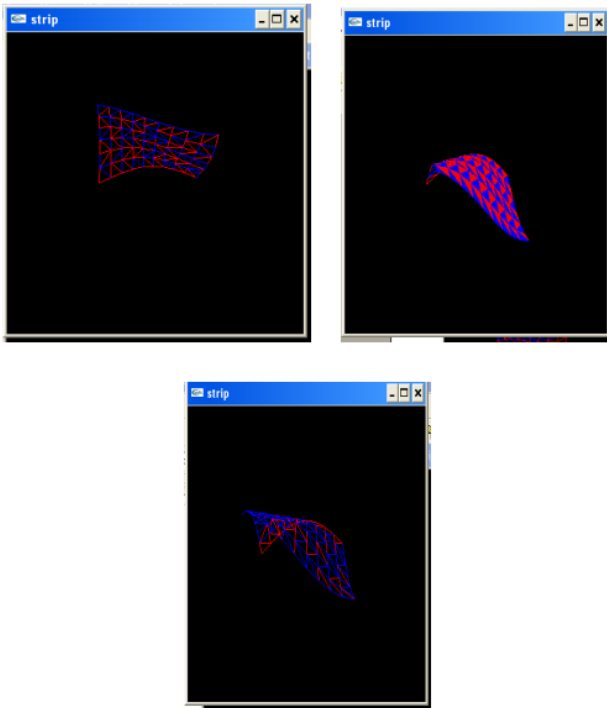


## SORU :

**SORU 1)** `GL_TRIANGLE_STRIP` kullanarak şekilde görüldüğü gibi üçgenlerden oluşan ve düzlemsel olmayan bir mesh çizdiriniz. Kullanıcı klavye aracılığıyla mesh 'i tersyüz edebilmelidir. Ayrıca klavyeden `f` tuşuna basıldığında meshin ön tarafını oluşturan poligonların içi o anki renk değeri ile doldurulmalıdır. Bunun için `glPolygonMode` fonksiyonunu kullanabilirsiniz.



## CEVAP :

main fonksiyonunda 10x10 matris değişken oluşturuldu.

```
for (k = 0; k < 10; k++) {  
    for (i = 0; i < 10; i++) {  
        mesh[k][i] = sin((float)(i+k)/10*3.1415)*3;  
    }  
}
```

döngüsü kullanılarak noktalar mesh değişkenine sinüs dalgası şekilde yazdırıldı.

glRotatef fonksiyonu ile x ve y de döndürme için tanımlandı. Döndürme açıları için rot\_x ve rot\_y değişkenleri kullanıldı. Tuş kontrol fonksiyonunda bu değişkenler kullanılarak ters düz etme işlemi tanımlandı.

```
void display(void) {  
    int i, k, swap = 0;  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
    glPushMatrix();  
    glTranslatef(-10/2, 0, -10*2);  
    glRotatef(rot_x, 0, 1, 0); //x döndürme  
    glRotatef(rot_y, 1, 0, 0); //y döndürme
```

Üçgenler gl\_triangle\_strip fonksiyonu kullanılarak çizdirilmiştir.

```
    glColor3f(255, 255, 255); //arka plan siyah  
    glBegin(GL_TRIANGLE_STRIP);  
    for (k = 0; k < 10-1; k++) {  
        if (swap) {  
            for (i = 10-1; i >= 0; i--) {  
                glColor3f(255, 0, 0);  
                glVertex3f(i, mesh[k][i], k);  
                glColor3f(0, 0, 255);  
                glVertex3f(i, mesh[k+1][i], k+1);  
                if (i == 0) //ilk üçgen mi degil mi check  
                    glVertex3f(i, mesh[k+1][i], k+1);  
            }  
        } else {  
            for (i = 0; i < 10; i++) {  
                glColor3f(255, 0, 0);  
                glVertex3f(i, mesh[k][i], k);  
                glColor3f(0, 0, 255);  
                glVertex3f(i, mesh[k+1][i], k+1);  
                if (i == 10-1) //son üçgen mi degil mi check  
                    glVertex3f(i, mesh[k+1][i], k+1);  
            }  
        }  
    }
```

```

swap ^= 1; //swap değişkenini sürekli tersliyor
} glEnd();
glPopMatrix();
glutSwapBuffers();}

```

#### Klavye tuş fonksiyonları

**Esc** Programı Kapatır

**w** Üst yüzeyin PolygonModunu Line ile Fill arası değiştirir.

**c** İlk konumdan 180 derece terse döndürür.

**r** Ters konumda 180 derece döndürerek ilk konuma getirir.

```

void keyboard(unsigned char key, int x, int y)
{
    if (key == 27) {
        exit(0);
    } else if (key == 'w' || key == 'W') { //Tuş kontrolü
        wire ^= 1; //her tuş basmada değeri tersler
        if (wireframe)
        { glPolygonMode(GL_FRONT, GL_LINE); //polygon mod line
        glutPostRedisplay(); }
        else
        { glPolygonMode(GL_FRONT, GL_FILL); //polygon mod fill
        glutPostRedisplay(); }
    } else if (key == 'c' || key == 'C') { //Tuş kontrolü
        rot_x = 180 - old_x; //glRotatef x 180 derece değiştirir
        rot_y = 180 - old_y; //glRotatef y 180 derece değiştirir
        glutPostRedisplay(); }
    else if (key == 'r' || key == 'R') { //Tuş kontrolü
        rot_x = old_x; //glRotatef x açısını ilk değere döndürür
        rot_y = old_y; //glRotatef y açısını ilk değere döndürür
        glutPostRedisplay();
    }
}

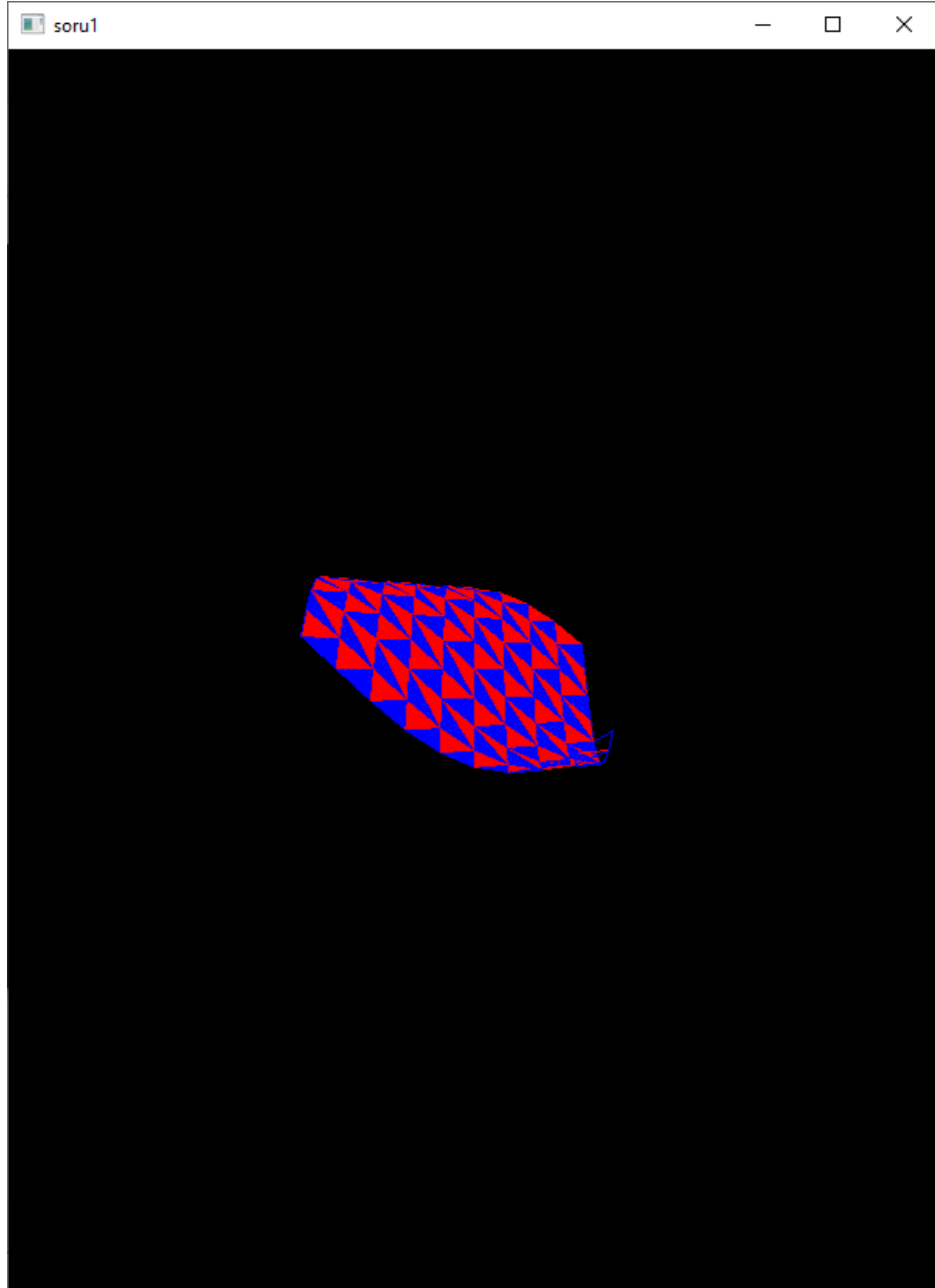
```

**Ek**

finalsoru1.cpp

finalsoru1.exe

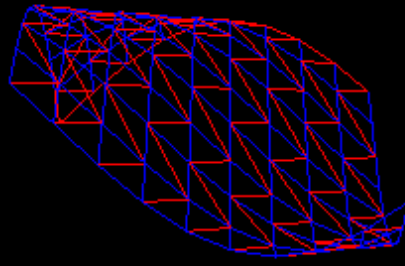
Ekran ıktısı



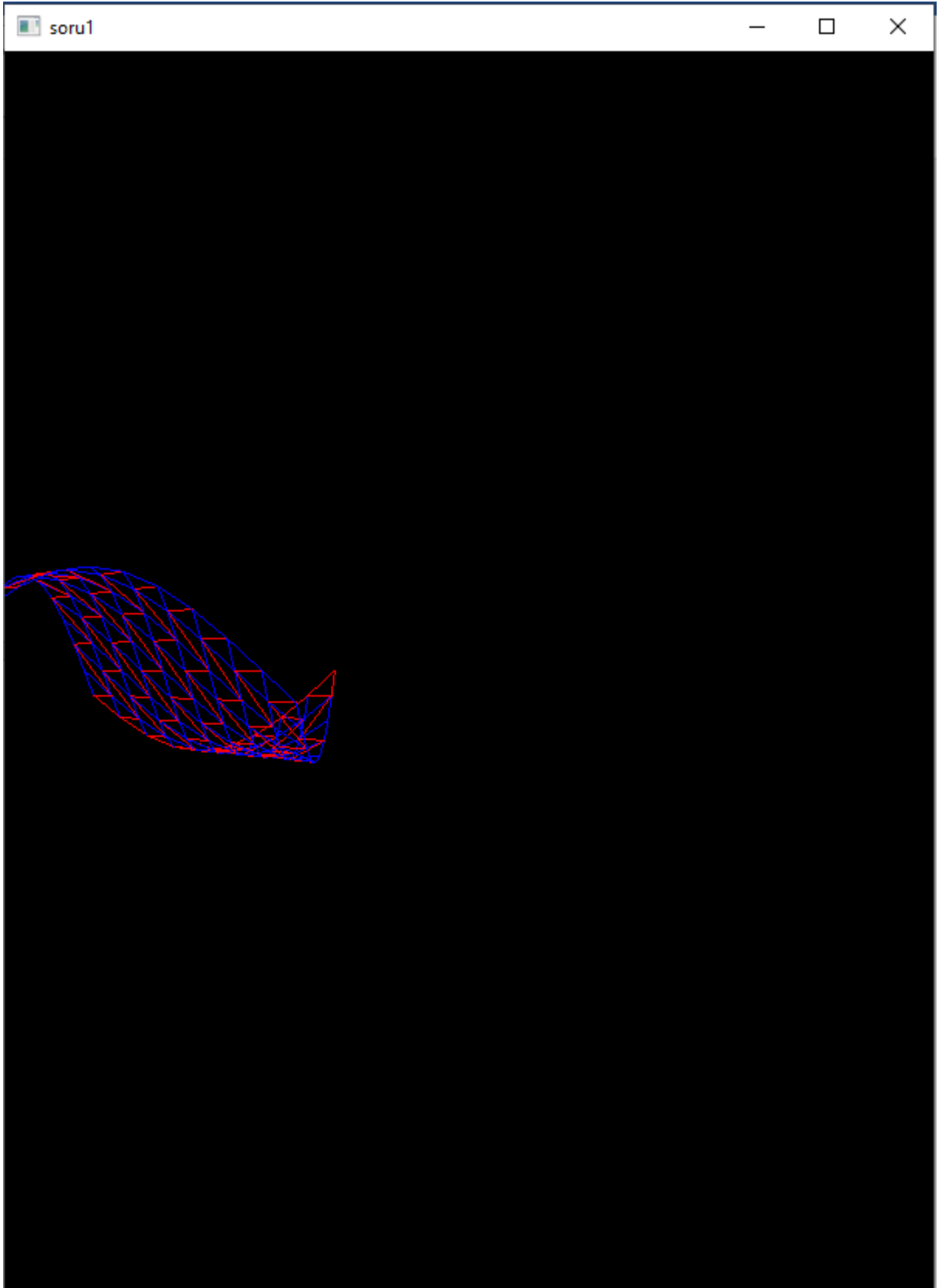
GL\_LINE aktif ekran ıktısı



soru1



180 derece döndürölmüş durum ekran çıktıı



180 derece döndürülmüş GL\_FILL aktif ekran çıktısı

