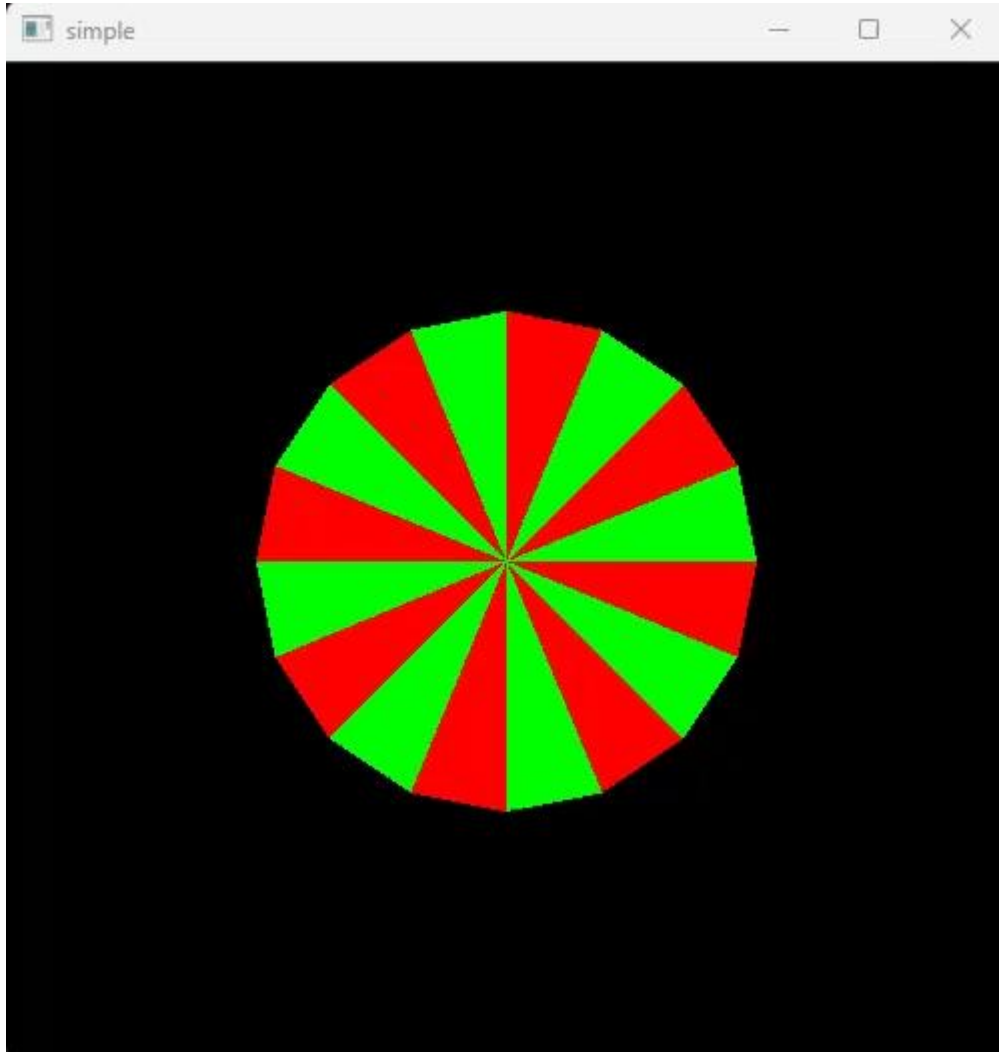
솔리드 물체(원뿔 그리기) (3_7_1)

Try

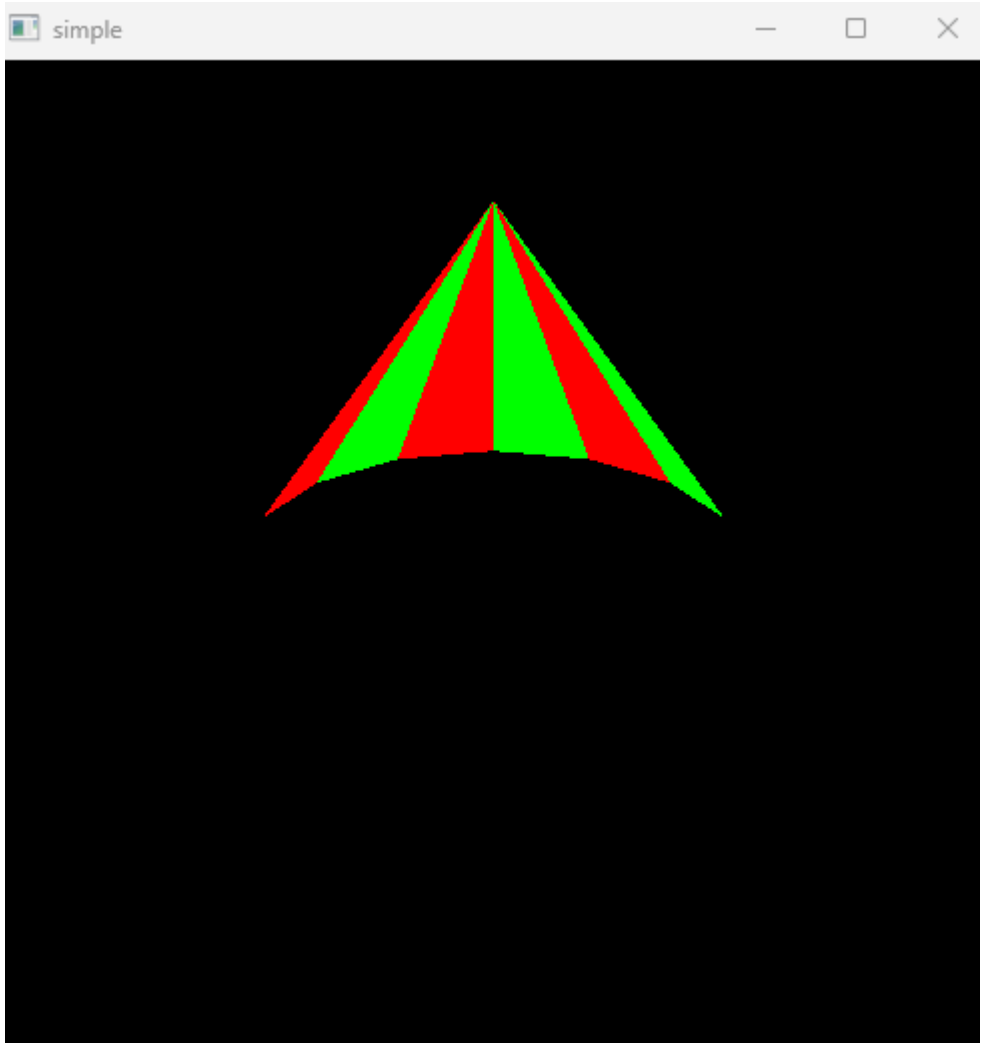
- **3_7_0 + 3_7_0_a 코드를 기반**으로 다음과 같은 원뿔을 그리시오

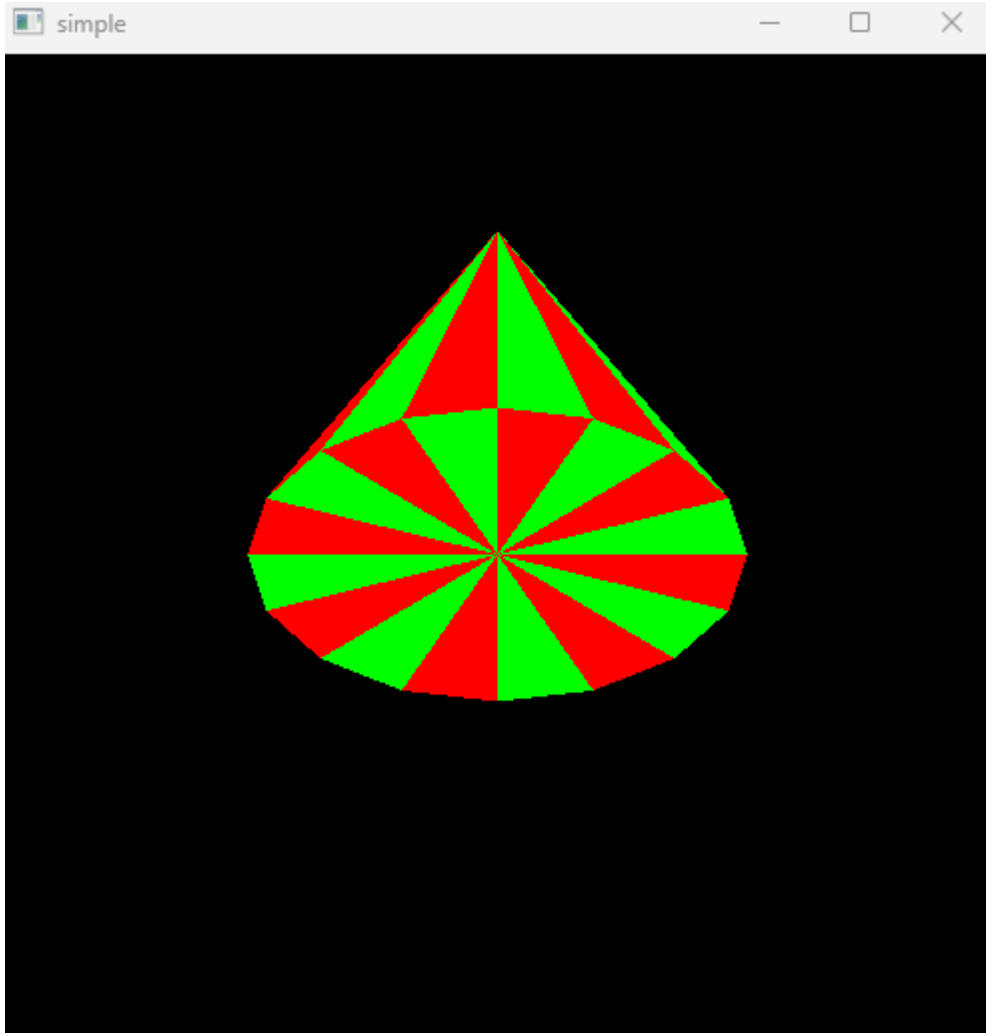


솔리드 물체(원뿔 그리기) (3_7_1)

Try

- 원뿔의 밑면을 보면 안 보임
- why?





솔리드 물체(원뿔 그리기) (3_7_1)

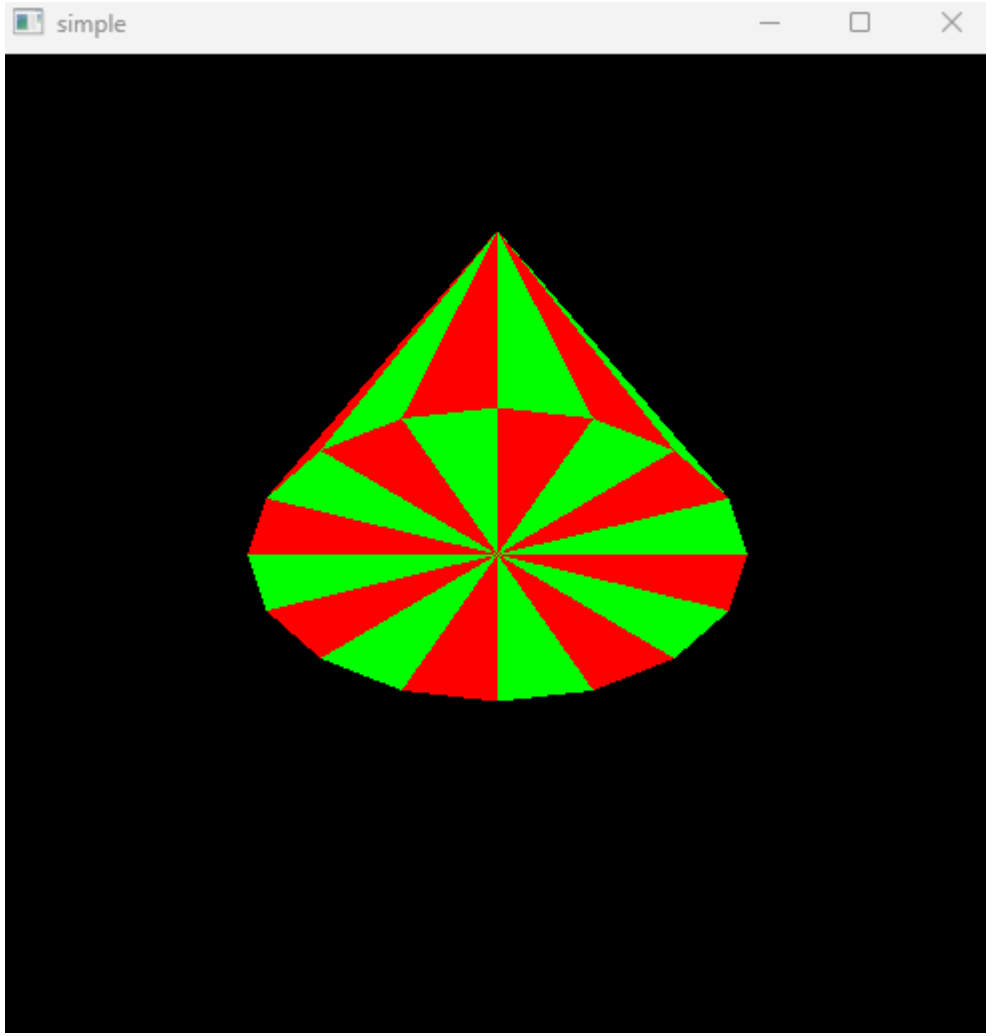
Try

- 원뿔의 밑면을 보면 안 보임
- why? 답
  - 밑면 원의 그리는 방향을 반대로..

```
GLfloat x, y, angle;
int iPivot = 1;
glBegin(GL_TRIANGLE_FAN);
glVertex2f(0.0f, 0.0f);
for (angle = 0.0f; angle < (2.0f * GL_PI); angle += (GL_PI / 8.0f))
{
    x = 50.0f * sin(angle);
    y = 50.0f * cos(angle);

    if ((iPivot % 2) == 0)
        glColor3f(0.0f, 1.0f, 0.0f);
    else
        glColor3f(1.0f, 0.0f, 0.0f);

    iPivot++;
    glVertex2f(x, y);
}
glEnd();
```



솔리드 물체(원뿔 그리기) (3_7_2)

- 깊이
  - 가리어진 면을 제거
  - 픽셀이 그려지면 깊이 값 이라는 값을 할당하여, 관측자로부터의 위치를 결정
  - 화면상의 같은 위치에 다른 픽셀이 그려질 때 깊이 값을 비교
  - **낮은 깊이 값을 가진 픽셀은 화면에 보임**

```
if (bDepth)
    glEnable(GL_DEPTH_TEST);
else
    glDisable(GL_DEPTH_TEST);
```

  - 이 모드를 사용하기 위해서는 depth 관련 기능을 enable해 줘야 함

```
//glClear(GL_COLOR_BUFFER_BIT);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

```
//glFlush();
glutSwapBuffers();
```

```
//glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
```

깊이

- glOrtho(left, right, bottom, top, near, far)는 클리핑 영역을 설정
  - near > far 이면 projection 계산 시 z축이 **뒤집혀서** 깊이 비교가 반대가 됨

- 카메라 뷰 기준으로 depth를 계산함
  - 그 depth의 기준은 glOrtho()에서 정한 near/far임.
    - ✓ near < far = 카메라에 가까이 있는 도형이 가까운 곳에 출력
    - ✓ near > far = 카메라에 가까이 있는 도형이 멀 곳에 출력
- 카메라 좌표는 투영될 화면 방향을 결정
  - gluLookAt(0, 0, 0, // 카메라 위치 (eye position)
    0.0f, 0.0f, -1, // 바라볼 목표 지점 (look-at point)
    0.0f, 1.0f, 0.0f); // 카메라 상향 벡터 (up direction) - 보통 (0, 1, 0) 사용

- 카메라가 원점에서 -z방향으로 보고 있고, ortho(…-100, 100)이고,
  xy평명에 같은 크기의 원A가 z=10, 원B가 z=20에 있다면?
  (후면제거 활성화, 원의 크기는 같음, GL_DEPTH_TEST 활성)
  - **어떤 원이 보일까???????**

깊이

- glOrtho(left, right, bottom, top, near, far)는 클리핑 영역을 설정
  - near > far 이면 projection 계산 시 z축이 **뒤집혀서** 깊이 비교가 반대가 됨

- 카메라 뷰 기준으로 depth를 계산함
  - 그 depth의 기준은 glOrtho()에서 정한 near/far임.
    - ✓ near < far = 카메라에 가까이 있는 도형이 가까운 곳에 출력
    - ✓ near > far = 카메라에 가까이 있는 도형이 멀 곳에 출력
- 카메라 좌표는 투영될 화면 방향을 결정
  - gluLookAt(0, 0, 0, // 카메라 위치 (eye position)
    0.0f, 0.0f, -1, // 바라볼 목표 지점 (look-at point)
    0.0f, 1.0f, 0.0f); // 카메라 상향 벡터 (up direction) - 보통 (0, 1, 0) 사용

- 카메라가 원점에서 -z방향으로 보고 있고, ortho(…-100, 100)이고,
  xy평명에 같은 크기의 원A가 z=10, 원B가 z=20에 있다면?
  (후면제거 활성화, 원의 크기는 같음, GL_DEPTH_TEST 활성)
  - **B**, 80점

깊이

- glOrtho(left, right, bottom, top, near, far)는 클리핑 영역을 설정
  - near > far 이면 projection 계산 시 z축이 **뒤집혀서** 깊이 비교가 반대가 됨

- 카메라 뷰 기준으로 depth를 계산함
  - 그 depth의 기준은 glOrtho()에서 정한 near/far임.
    - ✓ near < far = 카메라에 가까이 있는 도형이 가까운 곳에 출력
    - ✓ near > far = 카메라에 가까이 있는 도형이 멀 곳에 출력
- 카메라 좌표는 투영될 화면 방향을 결정
  - gluLookAt(0, 0, 0, // 카메라 위치 (eye position)
    0.0f, 0.0f, -1, // 바라볼 목표 지점 (look-at point)
    0.0f, 1.0f, 0.0f); // 카메라 상향 벡터 (up direction) - 보통 (0, 1, 0) 사용

- 카메라가 원점에서 -z방향으로 보고 있고, ortho(…-100, 100)이고,
  xy평명에 같은 크기의 원A가 z=10, 원B가 z=20에 있다면?
  (후면제거 활성화, 원의 크기는 같음, GL_DEPTH_TEST 활성)
  - **B, but 그려지는 방향 정보가 없으니 전면/후면 알 수 없음!**