```
/* simple2.c */
/* a simple openGL program for drawing lines, points and polygons */

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <assert.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

GLfloat red[3] = {1.0, 0.0, 0.0};
GLfloat green[3] = {0.0, 1.0, 0.0};
GLfloat blue[3] = {0.0, 0.0, 1.0};
GLfloat black[3] = {0.0, 0.0, 0.0};
GLfloat white[3] = {1.0, 1.0, 1.0};
GLfloat gray[3] = {0.5, 0.5, 0.5};
GLfloat yellow[3] = {1.0, 1.0, 0.0};
GLfloat magenta[3] = {1.0, 0.0, 1.0};
GLfloat cyan[3] = {0.0, 1.0, 1.0};

/* forward declarations of functions */
void display(void);
void keypress(unsigned char key, int x, int y);

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

  /* open a window */
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
  glutInitWindowSize(500, 500);
  glutInitWindowPosition(100,100);
  glutCreateWindow(argv[0]);

  /* OpenGL init */
  glClearColor(0, 0, 0, 0);   /* set background color black*/

  /* callback functions */
  glutDisplayFunc(display);
  glutKeyboardFunc(keypress);

  /* event handler */
  glutMainLoop();
  return 0;
}

void draw_polygon(){

  glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
  glColor3fv(blue);   /* set draw color blue */

 /* draw a polygon */
  glBegin(GL_POLYGON);
  glVertex2f(-0.5,-0.5);
  glVertex2f(-0.5,0.5);
  glVertex2f(0.5,0.5);
  glVertex2f(0.5,-0.5);
  glEnd();
}

void draw_rectangle() {
  glColor3fv(red);
  glRectf(0.0, 0.0, 0.2, 0.3);
}

void draw_lines() {
  glColor3fv(yellow);
  glBegin(GL_LINES);
  glVertex2f(-1.0,-1.0);
  glVertex2f(1.0,1.0);
  glVertex2f(-1.0,-0.8);
  glVertex2f(1.0,0.8);
  glEnd();
}

void draw_line_fan() {
  int i;
  float step=0.1, x0 = 0.3,y0=0.3, x1,x2,y1,y2;

  glColor3fv(green);
  for (i=0; i<10; i++) {
    x1 = x0+ i*step;
    y1 = y0;
    x2 = x0;
    y2 = y0+i*step;
     glBegin(GL_LINES);
     glVertex2f(x1,y1);
     glVertex2f(x2,y2);
     glEnd();
  }
}

void draw_points() {
  int i;
  glColor3fv(cyan);
  glBegin(GL_POINTS);
  for (i=0; i<10;i++) {
    glVertex2f(-0.5+i*0.03, 0.8);
  }
  glEnd();
}

void polar(GLfloat r, GLfloat fi, GLfloat* v) {
  v[0] = r * cos(fi);
  v[1] = r * sin(fi);
}

void display(void) {

  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

  draw_polygon();
  draw_rectangle();
  draw_lines();
  draw_line_fan();
  draw_points();

  /* execute the drawing commands */
  glFlush();
}

void keypress(unsigned char key, int x, int y) {
  switch(key) {
  case 'q':
    exit(0);
    break;
  }
}
```