

Q1

```
#include <string>

#include <stdlib.h>

#ifdef _APPLE_

#include <OpenGL/OpenGL.h>

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <iostream>

using namespace std;


void init() {

glClearColor (0.0, 0.0, 0.0, 0.0);

glShadeModel (GL_FLAT);

}


void display() {

int x,y,color=0;

glClear (GL_COLOR_BUFFER_BIT);

glColor3f (1.0, 0.0, 0.0);

for(x=1;x<=12;x++){

if(color==0){

glColor3f (1.0, 0.0, 0.0);

color++;

}

else{

glColor3f (1.0, 1.0, 1.0);

color=0;

}
```

```

}
for(y=1;y<=12;y++){
if(color==0){
glColor3f (1.0, 0.0, 0.0);
color++;
}
else{
glColor3f (1.0, 1.0, 1.0);
color=0;
}
glBegin(GL_QUADS);
glVertex2f(37.5+37.5*x, 37.5+37.5*y);
glVertex2f(37.5*x, 37.5+37.5*y);
glVertex2f(37.5*x, 37.5*y);
glVertex2f(37.5+37.5*x, 37.5*y);

glEnd();
}}
glFlush ();
}

void reshape (int w, int h) {
glViewport (0, 0, (GLsizei) w, (GLsizei) h);
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
gluOrtho2D (0.0, (GLdouble) w, 0.0, (GLdouble) h);
}

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

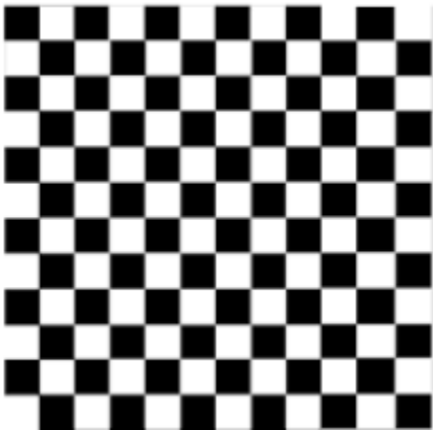
```

```

glutInit(&argc, argv);
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize (375, 375);
glutInitWindowPosition (100,100);
glutCreateWindow (argv[0]);
init ();
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutMainLoop();
return 0;

```

Output:



Q2

```

#include <windows.h> // for MS Windows
#include <GL/glut.h> // GLUT, include glu.h and gl.h
/* Initialize OpenGL Graphics */
void initGL(void) {
    // Set "clearing" or background color
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 1000, 0, 1000);
}

```

```

}

/* Handler for window-repaint event. Call back when the window first appears and
whenever the window needs to be re-painted. */

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT); // Clear the color buffer with current clearing color

    // Define shapes enclosed within a pair of glBegin and glEnd

    glBegin(GL_TRIANGLES);    // Each set of 3 vertices form a triangle

    glColor3f(1.0f, 0.0f, 0.0f); // Red

    glVertex2f(0, 10);
    glVertex2f(100, 10);
    glVertex2f(50, 100);

    glEnd();

    glScalef(2.0, 2.0, 2.0);

    glBegin(GL_TRIANGLES);    // Each set of 3 vertices form a triangle

    glColor3f(0.0f, 0.0f, 1.0f); // blue

    glVertex2f(200, 10);
    glVertex2f(300, 10);
    glVertex2f(250, 100);

    glEnd();

```

```

        glFlush(); // Render now

    }

/* Main function: GLUT runs as a console appGL_TRIANGLES    GL_TRIANGLE_STRIPlication starting at
main() */
int main(int argc, char** argv) {

    glutInit(&argc, argv); // Initialize GLUT
    glutCreateWindow("Program"); // Create window with the given title
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(1000, 1000); // Set the window's initial width & height
    glutInitWindowPosition(0, 0); // Position the window's initial top-left corner
    glutDisplayFunc(display); // Register callback handler for window re-paint event

    initGL(); // Our own OpenGL initialization
    glutMainLoop(); // Enter the event-processing loop

}

```

Output:

