COMPUTER GRAPHICS [B]

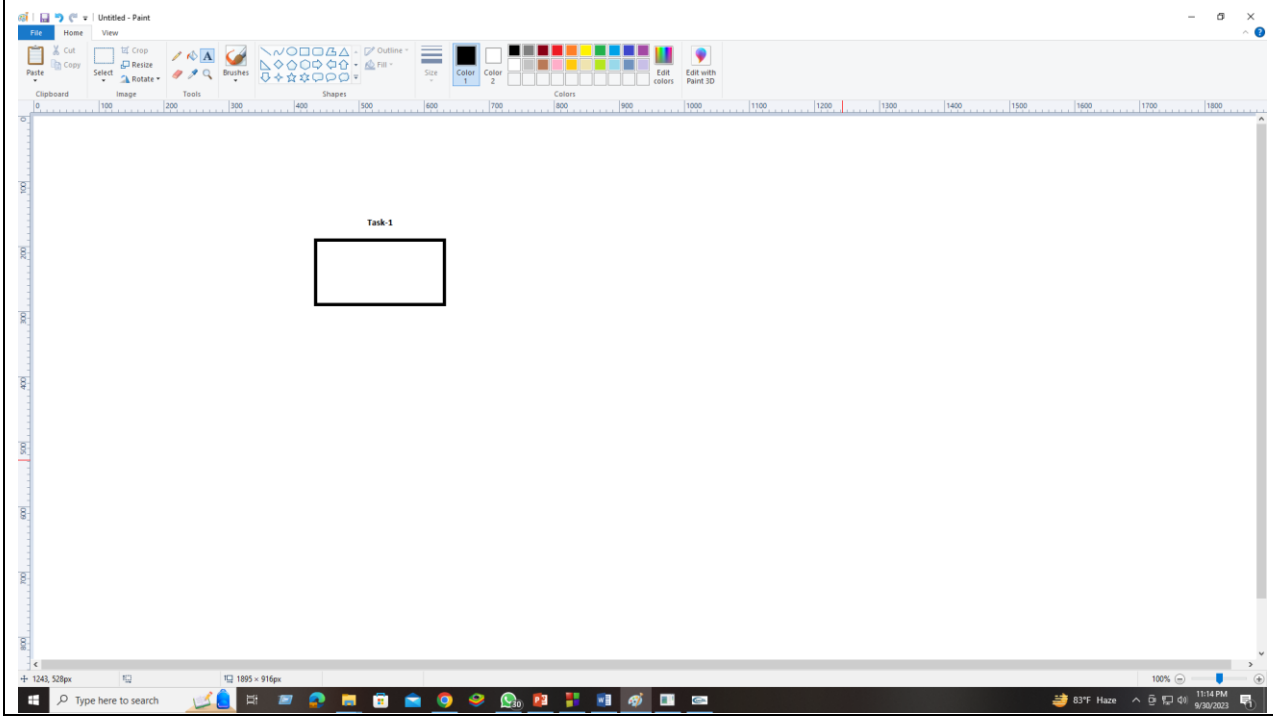
Lab Taks-1

Submission Guidelines-03.10.2023

Question-

Draw the object-

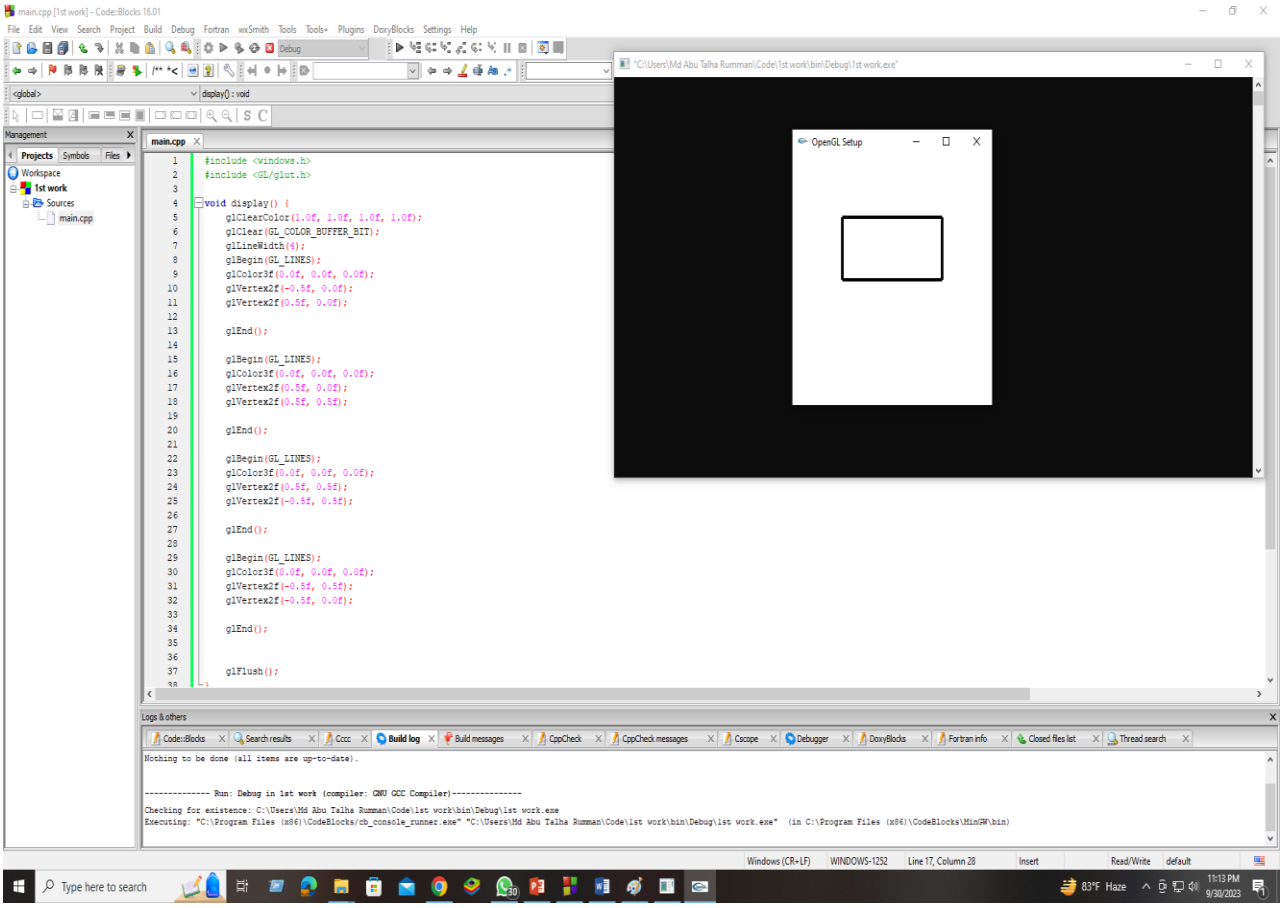
Graph Plot (Picture)-



Code-

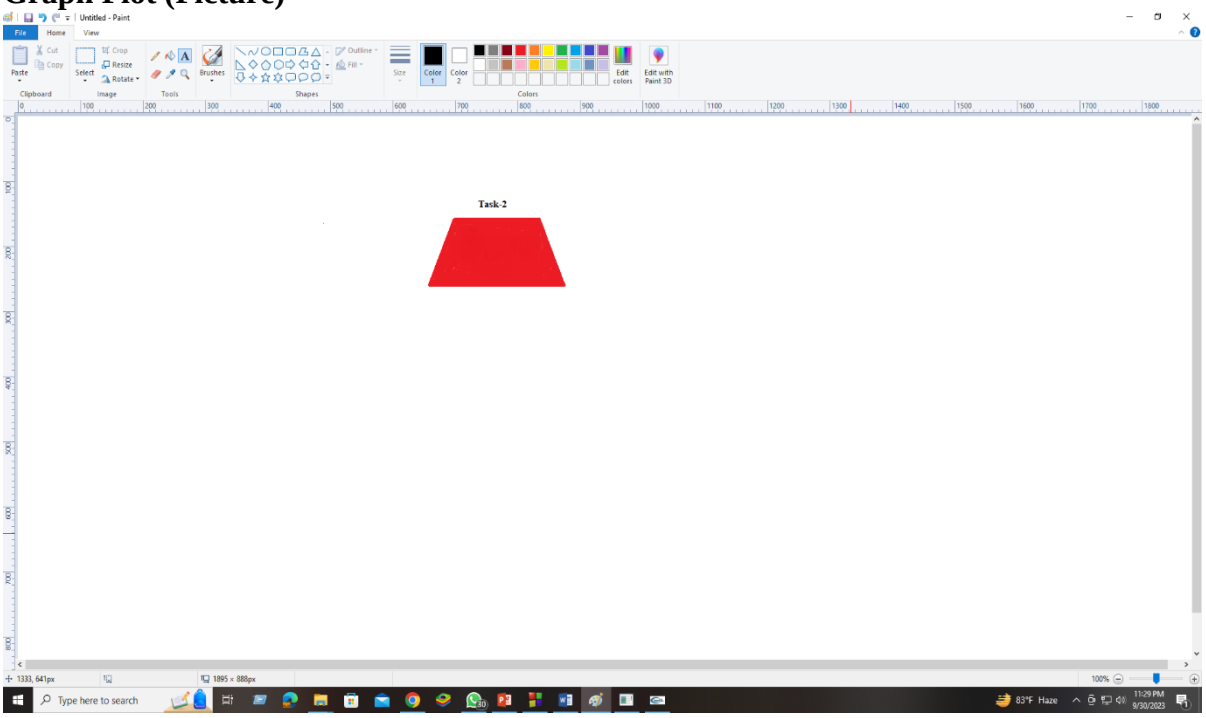
```
#include <windows.h>
#include <GL/glut.h>

void display() {
	glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
	glClear(GL_COLOR_BUFFER_BIT);
	glLineWidth(4);
	glBegin(GL_LINES);
	glColor3f(0.0f, 0.0f, 0.0f);
	glVertex2f(-0.5f, 0.0f);
	glVertex2f(0.5f, 0.0f);
	glEnd();

	glBegin(GL_LINES);
	glColor3f(0.0f, 0.0f, 0.0f);
	glVertex2f(0.5f, 0.0f);
	glVertex2f(0.5f, 0.5f);
	glEnd();

	glBegin(GL_LINES);
	glColor3f(0.0f, 0.0f, 0.0f);
	glVertex2f(0.5f, 0.5f);
	glVertex2f(-0.5f, 0.5f);
	glEnd();

	glBegin(GL_LINES);
	glColor3f(0.0f, 0.0f, 0.0f);
	glVertex2f(-0.5f, 0.5f);
	glVertex2f(-0.5f, 0.0f);

	glEnd();

	glFlush();
}

int main(int argc, char** argv)
 {
	glutInit(&argc, argv);
	glutCreateWindow("OpenGL Setup");
	glutInitWindowSize(320, 320);
	glutDisplayFunc(display);
	glutMainLoop();
	return 0;
}
```

Output Screenshot (Full Screen)-



Question-
Draw the object-

Graph Plot (Picture)-

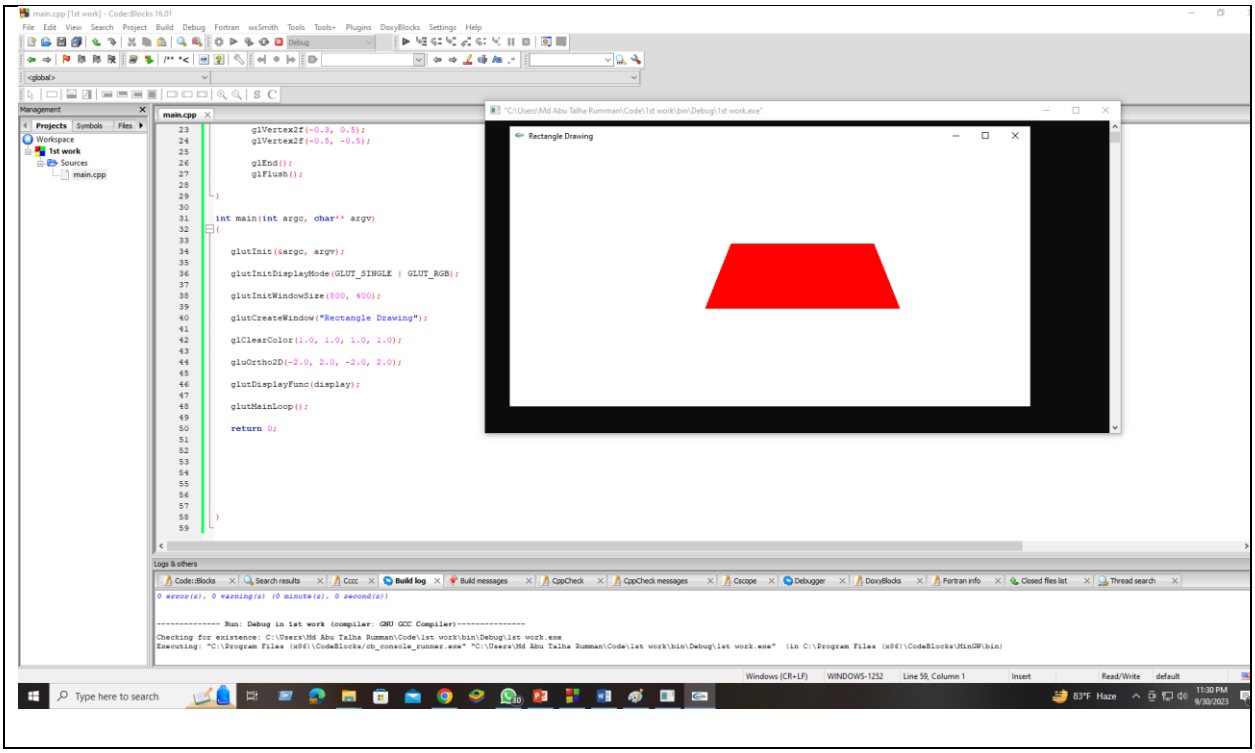


Code-

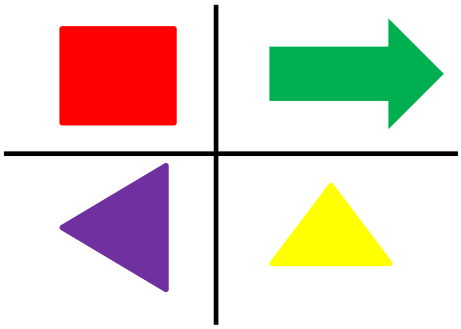
```
#include <windows.h>  // for MS Windows
#include <GL/glut.h>  // GLUT, include glu.h and gl.h
/* Initialize OpenGL Graphics */

void initGL() {

// Set "clearing" or background color

glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque
}
void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
      glColor3f(1.0, 0.0, 0.0);  // Set color to white
```

```
   glBegin(GL_POLYGON);
      glVertex2f(-0.5, -0.5);
      glVertex2f(1.0, -0.5);
      glVertex2f(1.0, -0.5);
      glVertex2f(0.8, 0.5);
      glVertex2f(0.8, 0.5);
      glVertex2f(-0.3, 0.5);
      glVertex2f(-0.3, 0.5);
      glVertex2f(-0.5, -0.5);
      glEnd();


      glFlush();

}

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

   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
   glutInitWindowSize(800, 400);
   glutCreateWindow("Rectangle Drawing");
   glClearColor(1.0, 1.0, 1.0, 1.0);
   gluOrtho2D(-2.0, 2.0, -2.0, 2.0);
   glutDisplayFunc(display);
   glutMainLoop();

   return 0;









}
```

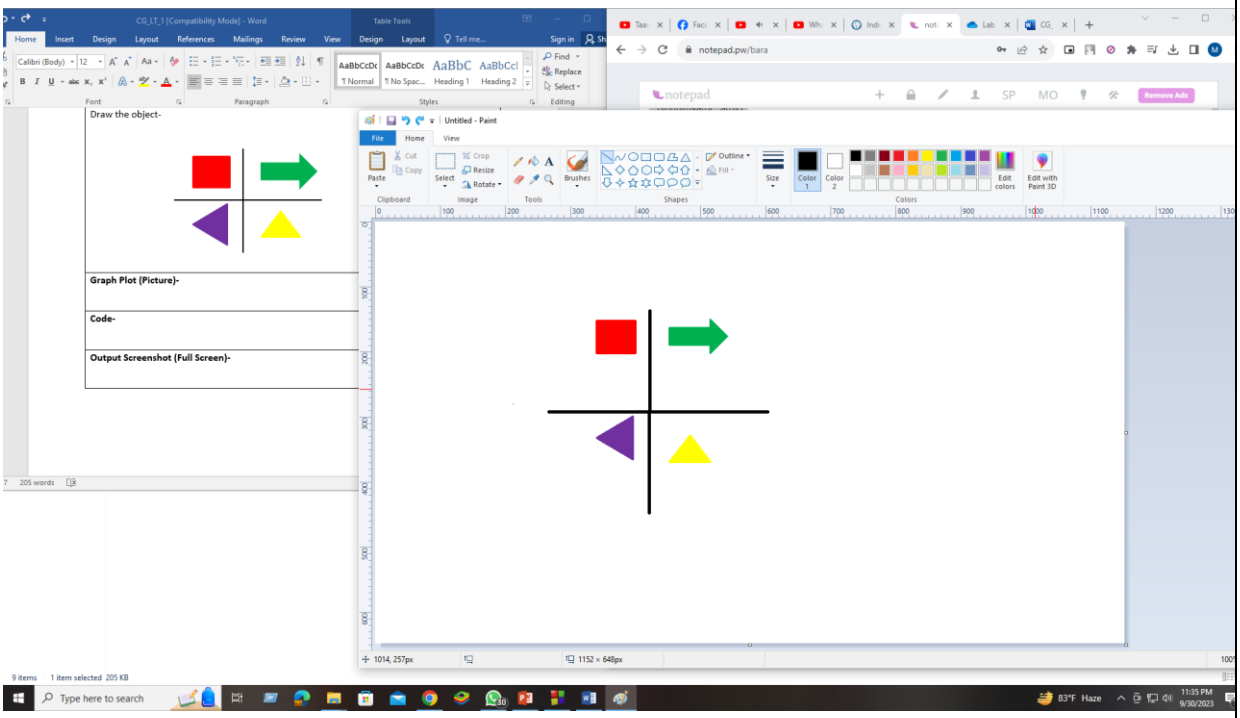
Output Screenshot (Full Screen)-


main.cpp [1st work] - Code::Blocks 16.01
Rectangle Drawing
Type here to search

Question-
Draw the object-



Graph Plot (Picture)-



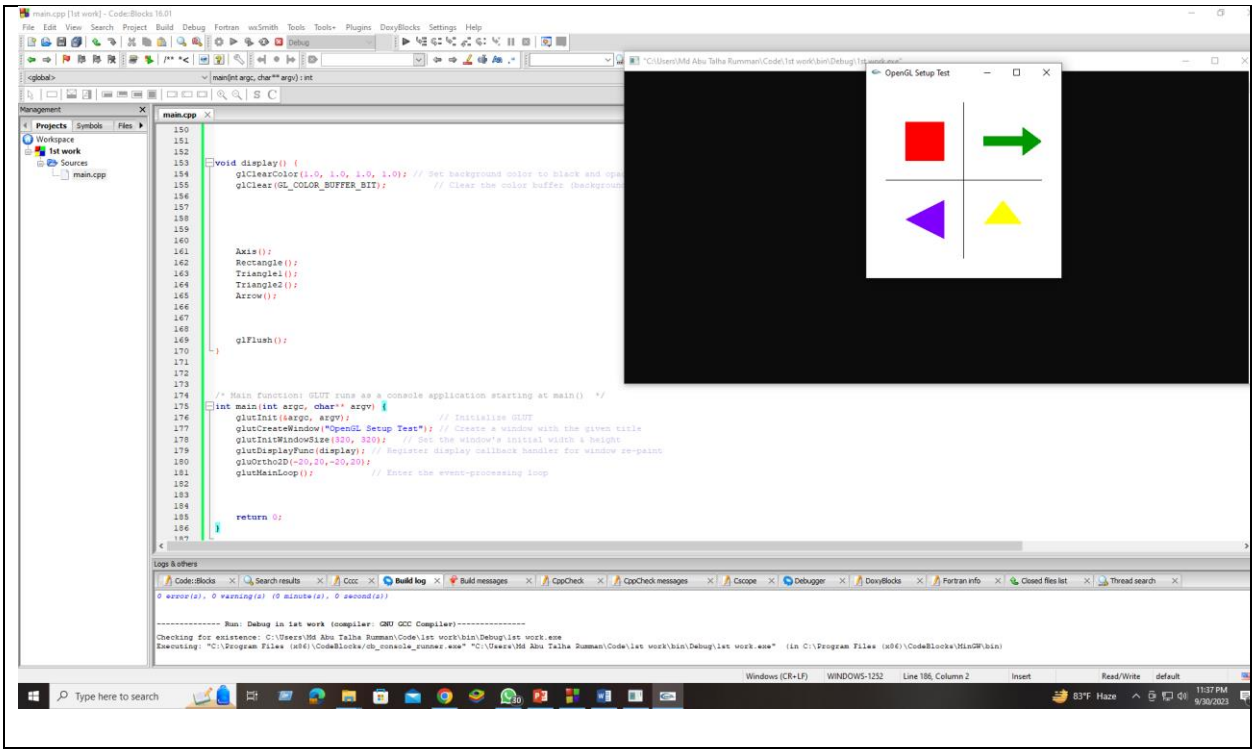
Code-

```
#include <windows.h>  // for MS Windows
#include <GL/glut.h>  // GLUT, include glu.h and gl.h

void Axis()
{

   glBegin(GL_LINES);
        glColor3f(0.0f, 0.0f, 0.0f); // Black color
```

```
        glVertex2f(-16.0, 0.0);
        glVertex2f(16.0, 0.0);
        glVertex2f(0.0, 16.0);
        glVertex2f(0.0, -16.0);
        glEnd();

}

void Rectangle(){

   glBegin(GL_POLYGON);
        glColor3f(1.0f, 0.0f, 0.0f);
         glVertex2f(-12.0, 4.0);
        glVertex2f(-4.0, 4.0);
        glVertex2f(-4.0, 12.0);
        glVertex2f(-12.0, 12.0);
        glEnd();
}

void Triangle1(){
   glBegin(GL_POLYGON);
        glColor3ub(127, 0.0, 255);
        glVertex2f(-4.0, -12.0);
        glVertex2f(-4.0, -4.0);
        glVertex2f(-12.0,-8.0);
        glEnd();

}


void Triangle2(){

   glBegin(GL_POLYGON);
        glColor3ub(255, 255, 0);
        glVertex2f(4.0, -9.0);
        glVertex2f(12.0, -9.0);
        glVertex2f(8.0,-4.0);
        glEnd();



}

void Arrow(){
   glBegin(GL_POLYGON);
        glColor3ub(0, 153, 0);
```

```
        glVertex2f(4.0, 7.0);
        glVertex2f(12.0, 7.0);
        glVertex2f(12.0, 9.0);
        glVertex2f(4.0, 9.0)
        glEnd();

   glBegin(GL_POLYGON);
        glColor3ub(0, 153, 0);
        glVertex2f(12.0, 5.0);
        glVertex2f(16.0, 8.0);
        glVertex2f(12.0, 11.0);
        glEnd();

}

void display() {
        glClearColor(1.0, 1.0, 1.0, 1.0); // Set background color to black and opaque
        glClear(GL_COLOR_BUFFER_BIT);        // Clear the color buffer (background)

   Axis();
   Rectangle();
   Triangle1();
   Triangle2();
   Arrow();

        glFlush();
}

/* Main function: GLUT runs as a console application starting at main()  */
int main(int argc, char** argv) {
        glutInit(&argc, argv);              // Initialize GLUT
        glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title
        glutInitWindowSize(320, 320);   // Set the window's initial width & height
        glutDisplayFunc(display); // Register display callback handler for window re-paint
        gluOrtho2D(-20,20,-20,20);
        glutMainLoop();          // Enter the event-processing loop



        return 0;
}
```

Output Screenshot (Full Screen)-


OpenGL Setup Test
void display() {
glClearColor(1.0, 1.0, 1.0, 1.0);
glClear(GL_COLOR_BUFFER_BIT);
Axis();
Rectangle();
Triangle1();
Triangle2();
Arrow();
glFlush();
}
/* Main function: GLUT runs as a console application starting at main() */
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutCreateWindow("OpenGL Setup Test");
glutInitWindowSize(320, 320);
glutDisplayFunc(display);
gluOrtho2D(-20,20,-20,20);
glutMainLoop();
return 0;
}