Jan 23

TEMPLATE CODE

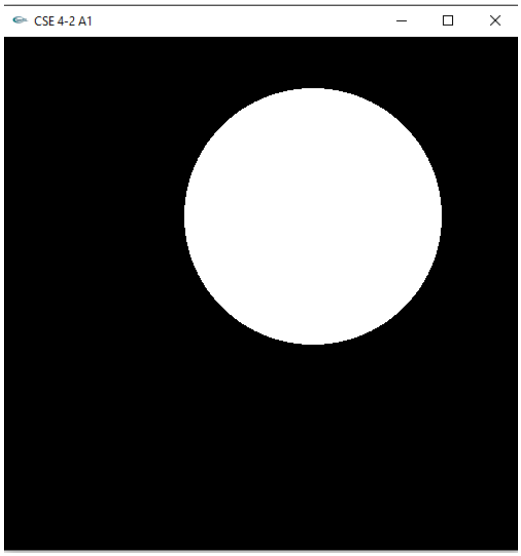
==========

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

void display(void)
{


}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowSize(500, 500);
   glutInitWindowPosition(100,100);
   glutCreateWindow("CSE 4-2 B2");
   glutDisplayFunc(display);
   glutMainLoop();
   return 0;
}
```

Jan 23

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

void display(void)
{
   /*glBegin(GL_POLYGON);
   glVertex2f(0, 0.6);
   glVertex2f(0.6, 0.2);
   glVertex2f(0.4, -0.5);
   glVertex2f(-0.4, -0.5);
   glVertex2f(-0.6, 0.2);
   glEnd();
   glFlush();*/


   /*glBegin(GL_POINTS);

   for(float i=-1; i<=1; i=i+0.002)
   {
```

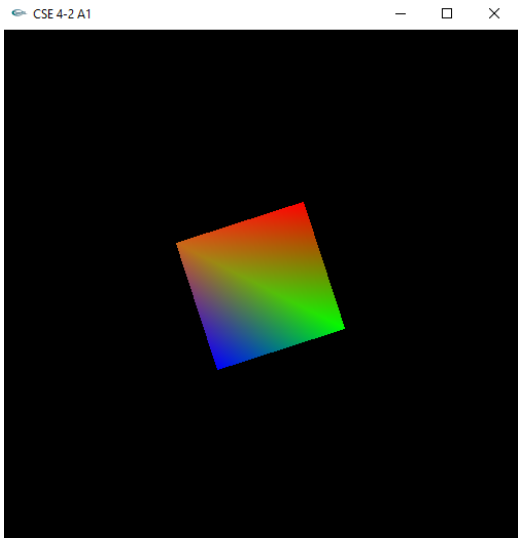
```
        glVertex2f(i, i); //this line will execute 1000 times.
    }

    glEnd();
    glFlush();*/


    glBegin(GL_POLYGON);
    float r = 0.5;
    for(float theta = 0; theta < 360; theta = theta + 0.01)
    {
        float x = 0.2 + r*cos(theta);
        float y = 0.3 + r*sin(theta);
        glVertex2f(x, y);
    }

    glEnd();
    glFlush();

}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("CSE 4-2 A1");
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```



Jan 30

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

void display(void)
{
   //glTranslatef(-0.4, -0.4, 0);
   //glRotatef(-45, 0, 0, 1);
   //glScalef(0.5, 0.5, 1);

   glClear(GL_COLOR_BUFFER_BIT);
   glBegin(GL_QUADS);
   glColor3f(1, 0, 0);
   glVertex2f(0.2, 0.2);
   glColor3f(0, 1, 0);
   glVertex2f(0.2, -0.2);
   glColor3f(0, 0, 1);
   glVertex2f(-0.2, -0.2);
   glColor3f(0.8, 0.4, 0.1);
   glVertex2f(-0.2, 0.2);
   glEnd();
   glFlush();
}

void keyboard(int key, int x, int y)
{
   switch(key)
   {
   case GLUT_KEY_UP:
      glTranslatef(0, 0.01, 0);
      glutPostRedisplay();
      break;
   case GLUT_KEY_DOWN:
      glTranslatef(0, -0.01, 0);
      glutPostRedisplay();
      break;
   case GLUT_KEY_RIGHT:
      glTranslatef(0.01, 0, 0);
      glutPostRedisplay();
      break;
   case GLUT_KEY_LEFT:
      glTranslatef(-0.01, 0, 0);
      glutPostRedisplay();
      break;
   }
```

```
}
void keypress(unsigned char key, int x, int y)
{
   switch(key)
   {
   case 'r':
      glRotatef(1, 0, 0, 1);
      glutPostRedisplay();
      break;
   case 'c':
      glRotatef(-1, 0, 0, 1);
      glutPostRedisplay();
      break;
   case 'm':
      glScalef(1.1, 1.1, 1);
      glutPostRedisplay();
      break;
   case 'n':
      glScalef(0.9, 0.9, 1);
      glutPostRedisplay();
      break;
   }
}
int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowSize(500, 500);
   glutInitWindowPosition(100,100);
   glutCreateWindow("CSE 4-2 A1");
   glutDisplayFunc(display);
   glutSpecialFunc(keyboard);
   glutKeyboardFunc(keypress);
   glutMainLoop();
   return 0;
}
```



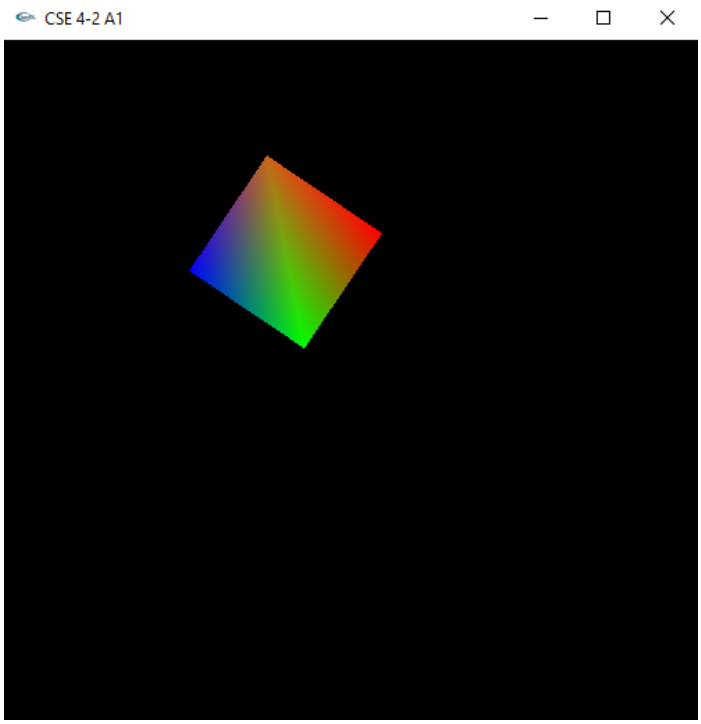
Feb 6

```
#include <windows.h>
#include <GL/glut.h>
#include <math.h>
#define pi 3.14159265
#define deg 45

float xa=0.2, ya=0.2, xb=0.2, yb=-0.2, xc=-0.2, yc=-0.2, xd=-0.2, yd=0.2;
float temp;
void display(void)
{
   //glTranslatef(-0.4, -0.4, 0);
   //glRotatef(-45, 0, 0, 1);
   //glScalef(0.5, 0.5, 1);

   glClear(GL_COLOR_BUFFER_BIT);
   glBegin(GL_QUADS);
   glColor3f(1, 0, 0);
   glVertex2f(xa, ya);
   glColor3f(0, 1, 0);
   glVertex2f(xb, yb);
   glColor3f(0, 0, 1);
   glVertex2f(xc, yc);
   glColor3f(0.8, 0.4, 0.1);
   glVertex2f(xd, yd);
   glEnd();
   glFlush();
}

void keyboard(int key, int x, int y)
```

```
{
   switch(key)
   {
   case GLUT_KEY_UP:
      //glTranslatef(0, 0.01, 0);
      ya = ya + 0.01;
      yb = yb + 0.01;
      yc = yc + 0.01;
      yd = yd + 0.01;
      glutPostRedisplay();
      break;
   case GLUT_KEY_DOWN:
      //glTranslatef(0, -0.01, 0);
      ya = ya - 0.01;
      yb = yb - 0.01;
      yc = yc - 0.01;
      yd = yd - 0.01;
      glutPostRedisplay();
      break;
   case GLUT_KEY_RIGHT:
      //glTranslatef(0.01, 0, 0);
      xa = xa + 0.01;
      xb = xb + 0.01;
      xc = xc + 0.01;
      xd = xd + 0.01;
      glutPostRedisplay();
      break;
   case GLUT_KEY_LEFT:
      //glTranslatef(-0.01, 0, 0);
      xa = xa - 0.01;
      xb = xb - 0.01;
      xc = xc - 0.01;
      xd = xd - 0.01;
      glutPostRedisplay();
      break;
   }
}

void keypress(unsigned char key, int x, int y)
{
   switch(key)
   {
   case 'r':
      //glRotatef(1, 0, 0, 1);
      temp = xa;
      xa = xa*cos(pi/180) - ya*sin(pi/180);
      ya = temp*sin(pi/180) + ya*cos(pi/180);
```

```
    temp = xb;
    xb = xb*cos(pi/180) - yb*sin(pi/180);
    yb = temp*sin(pi/180) + yb*cos(pi/180);

    temp = xc;
    xc = xc*cos(pi/180) - yc*sin(pi/180);
    yc = temp*sin(pi/180) + yc*cos(pi/180);

    temp = xd;
    xd = xd*cos(pi/180) - yd*sin(pi/180);
    yd = temp*sin(pi/180) + yd*cos(pi/180);
    glutPostRedisplay();
    break;
  case 'c':
    //glRotatef(-1, 0, 0, 1);
    temp = xa;
    xa = xa*cos(-pi/180) - ya*sin(-pi/180);
    ya = temp*sin(-pi/180) + ya*cos(-pi/180);

    temp = xb;
    xb = xb*cos(-pi/180) - yb*sin(-pi/180);
    yb = temp*sin(-pi/180) + yb*cos(-pi/180);

    temp = xc;
    xc = xc*cos(-pi/180) - yc*sin(-pi/180);
    yc = temp*sin(-pi/180) + yc*cos(-pi/180);

    temp = xd;
    xd = xd*cos(-pi/180) - yd*sin(-pi/180);
    yd = temp*sin(-pi/180) + yd*cos(-pi/180);
    glutPostRedisplay();
    break;
  case 'm':
    //glScalef(1.1, 1.1, 1);
    xa = xa*1.1;
    xb = xb*1.1;
    xc = xc*1.1;
    xd = xd*1.1;
    ya = ya*1.1;
    yb = yb*1.1;
    yc = yc*1.1;
    yd = yd*1.1;
    glutPostRedisplay();
    break;
  case 'n':
    //glScalef(0.9, 0.9, 1);
```

```
        xa = xa*0.9;
        xb = xb*0.9;
        xc = xc*0.9;
        xd = xd*0.9;
        ya = ya*0.9;
        yb = yb*0.9;
        yc = yc*0.9;
        yd = yd*0.9;
        glutPostRedisplay();
        break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("CSE 4-2 A1");
    glutDisplayFunc(display);
    glutSpecialFunc(keyboard);
    glutKeyboardFunc(keypress);
    glutMainLoop();
    return 0;
}
```



Feb 13

```
#include <windows.h>
#include <GL/glut.h>
#include <math.h>
#include <stdio.h>
#define pi 3.14159265
#define deg 45

float xa=0.2, ya=0.2, xb=0.2, yb=-0.2, xc=-0.2, yc=-0.2, xd=-0.2, yd=0.2;
float temp;
void display(void)
{
   //glTranslatef(-0.4, -0.4, 0);
   //glRotatef(-45, 0, 0, 1);
   //glScalef(0.5, 0.5, 1);

   glClear(GL_COLOR_BUFFER_BIT);
   glBegin(GL_QUADS);

   glColor3f(0, 1, 0);
   glVertex2f(0.4, 0.4);
   glVertex2f(0.4, -0.4);
   glVertex2f(-0.4, -0.4);
   glVertex2f(-0.4, 0.4);


   glColor3f(1, 0, 0);
   glVertex2f(xa, ya);
   glVertex2f(xb, yb);
   glVertex2f(xc, yc);
   glVertex2f(xd, yd);

   glEnd();
   glFlush();
}

void keyboard(int key, int x, int y)
{
   switch(key)
   {
   case GLUT_KEY_UP:
      //glTranslatef(0, 0.01, 0);
      ya = ya + 0.01;
      yb = yb + 0.01;
```

```
        yc = yc + 0.01;
        yd = yd + 0.01;
        glutPostRedisplay();

        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
           printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
           printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
           printf("Left ");
        }

        printf("\n");

        break;
    case GLUT_KEY_DOWN:
        //glTranslatef(0, -0.01, 0);
        ya = ya - 0.01;
        yb = yb - 0.01;
        yc = yc - 0.01;
```

```
        yd = yd - 0.01;
        glutPostRedisplay();
        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
           printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
           printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
           printf("Left ");
        }

        printf("\n");
        break;
    case GLUT_KEY_RIGHT:
        //glTranslatef(0.01, 0, 0);
        xa = xa + 0.01;
        xb = xb + 0.01;
        xc = xc + 0.01;
        xd = xd + 0.01;
        glutPostRedisplay();
        //inside
```

```
      if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
      {
        printf("Inside ");
      }

      //up
      if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
      {
         printf("Up ");
      }

      //down
      if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
      {
         printf("Down ");
      }

      //right
      if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
      {
         printf("Right ");
      }

      //Left
      if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
      {
         printf("Left ");
      }

      printf("\n");
      break;
   case GLUT_KEY_LEFT:
      //glTranslatef(-0.01, 0, 0);
      xa = xa - 0.01;
      xb = xb - 0.01;
      xc = xc - 0.01;
      xd = xd - 0.01;
      glutPostRedisplay();
      //inside
      if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
```

```
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
           printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
           printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
           printf("Left ");
        }

        printf("\n");
        break;
    }
}

void keypress(unsigned char key, int x, int y)
{
    switch(key)
    {
    case 'r':
        //glRotatef(1, 0, 0, 1);
        temp = xa;
        xa = xa*cos(pi/180) - ya*sin(pi/180);
        ya = temp*sin(pi/180) + ya*cos(pi/180);

        temp = xb;
```

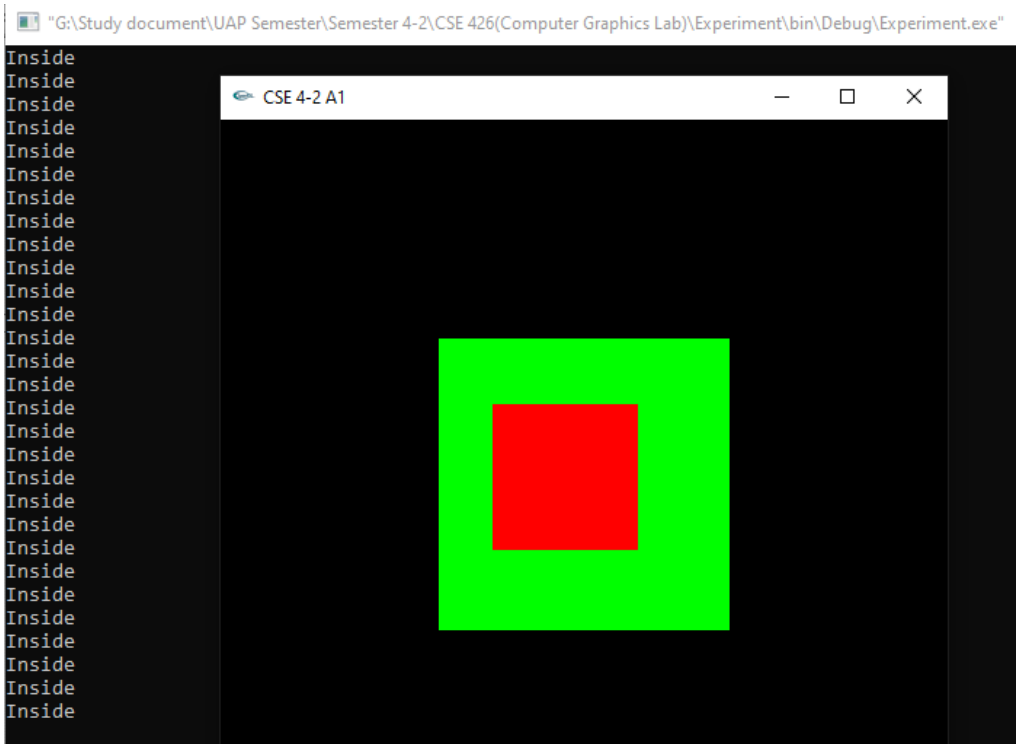
```
        xb = xb*cos(pi/180) - yb*sin(pi/180);
        yb = temp*sin(pi/180) + yb*cos(pi/180);

        temp = xc;
        xc = xc*cos(pi/180) - yc*sin(pi/180);
        yc = temp*sin(pi/180) + yc*cos(pi/180);

        temp = xd;
        xd = xd*cos(pi/180) - yd*sin(pi/180);
        yd = temp*sin(pi/180) + yd*cos(pi/180);
        glutPostRedisplay();

        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
           printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
           printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
           printf("Left ");
        }
```

```
        printf("\n");

        break;
    case 'c':
        //glRotatef(-1, 0, 0, 1);
        temp = xa;
        xa = xa*cos(-pi/180) - ya*sin(-pi/180);
        ya = temp*sin(-pi/180) + ya*cos(-pi/180);

        temp = xb;
        xb = xb*cos(-pi/180) - yb*sin(-pi/180);
        yb = temp*sin(-pi/180) + yb*cos(-pi/180);

        temp = xc;
        xc = xc*cos(-pi/180) - yc*sin(-pi/180);
        yc = temp*sin(-pi/180) + yc*cos(-pi/180);

        temp = xd;
        xd = xd*cos(-pi/180) - yd*sin(-pi/180);
        yd = temp*sin(-pi/180) + yd*cos(-pi/180);
        glutPostRedisplay();

        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
           printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
```

```
        {
            printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
            printf("Left ");
        }

        printf("\n");

        break;
    case 'm':
        //glScalef(1.1, 1.1, 1);
        xa = xa*1.1;
        xb = xb*1.1;
        xc = xc*1.1;
        xd = xd*1.1;
        ya = ya*1.1;
        yb = yb*1.1;
        yc = yc*1.1;
        yd = yd*1.1;
        glutPostRedisplay();

        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
           printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
        {
            printf("Down ");
        }
```

```
        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
           printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
           printf("Left ");
        }

        printf("\n");

        break;
    case 'n':
        //glScalef(0.9, 0.9, 1);
        xa = xa*0.9;
        xb = xb*0.9;
        xc = xc*0.9;
        xd = xd*0.9;
        ya = ya*0.9;
        yb = yb*0.9;
        yc = yc*0.9;
        yd = yd*0.9;
        glutPostRedisplay();

        //inside
        if(xa>=-0.4 && xa<=0.4 && xb>=-0.4 && xb<=0.4 && xc>=-0.4 && xc<=0.4 && xd>=-0.4 &&
xd<=0.4 && ya>=-0.4 && ya<=0.4 && yb>=-0.4 && yb<=0.4 && yc>=-0.4 && yc<=0.4 &&
yd>=-0.4 && yd<=0.4)
        {
          printf("Inside ");
        }

        //up
        if(ya>0.4 && yb>0.4 && yb>0.4 && yd>0.4)
        {
           printf("Up ");
        }

        //down
        if(ya<-0.4 && yb<-0.4 && yb<-0.4 && yd<-0.4)
```

```
        {
            printf("Down ");
        }

        //right
        if(xa>0.4 && xb>0.4 && xb>0.4 && xd>0.4)
        {
            printf("Right ");
        }

        //Left
        if(xa<-0.4 && xb<-0.4 && xb<-0.4 && xd<-0.4)
        {
            printf("Left ");
        }

        printf("\n");

        break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("CSE 4-2 A1");
    glutDisplayFunc(display);
    glutSpecialFunc(keyboard);
    glutKeyboardFunc(keypress);
    glutMainLoop();
    return 0;
}
```



Lab Task 2

```
#include <windows.h>
#include <GL/glut.h>
#include <math.h>
#include <stdio.h>
#define pi 3.14159265

float xa=0.2, ya=0.2, xb=0.2, yb=-0.2, xc=-0.2, yc=-0.2, xd=-0.2, yd=0.2;
void display(void)
{
   //glTranslatef(-0.4, 0.4, 0);

   //glRotatef(-45, 0, 0, 1);

   //glScalef(0.5, 0.5, 1);

   glClear(GL_COLOR_BUFFER_BIT);

   glBegin(GL_QUADS);

   glColor3f(0, 1, 0);
   glVertex2f(0.5, 0.5);
   glVertex2f(0.5, -0.5);
```

```
    glVertex2f(-0.5, -0.5);
    glVertex2f(-0.5, 0.5);

    glColor3f(1, 0, 0);
    glVertex2f(xa, ya);

    //glColor3f(0, 1, 0);
    glVertex2f(xb, yb);

    //glColor3f(0, 0, 1);
    glVertex2f(xc, yc);

    //glColor3f(0.6, 0.4, 0.1);
    glVertex2f(xd, yd);

    glEnd();
    glFlush();

}

void keyboard(unsigned char key, int x, int y)
{
    switch(key)
    {
    case 'w':
        //glTranslatef(0, 0.01, 0);
        ya = ya + 0.01;
        yb = yb + 0.01;
        yc = yc + 0.01;
        yd = yd + 0.01;
        glutPostRedisplay();

        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }
        else if((xa>=-0.5 && xa<=0.5 && ya>=-0.5 && ya<=0.5) || (xb>=-0.5 && xb<=0.5 &&
yb>=-0.5 && yb<=0.5) || (xc>=-0.5 && xc<=0.5 && yc>=-0.5 && yc<=0.5) || (xd>=-0.5 &&
xd<=0.5 && yd>=-0.5 && yd<=0.5))
        {
          printf("Intersect with ");
```

```
        }

        //up
        if(ya>0.5 || yb>0.5 || yc>0.5 || yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 || yb<-0.5 || yc<-0.5 || yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 || xb>0.5 || xc>0.5 || xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 || xb<-0.5 || xc<-0.5 || xd<-0.5)
        {
            printf("Left ");
        }

        printf("\n");

        break;

    case 's':
        //glTranslatef(0, -0.01, 0);
        ya = ya - 0.01;
        yb = yb - 0.01;
        yc = yc - 0.01;
        yd = yd - 0.01;
        glutPostRedisplay();
        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }
```

```
        else if((xa>=-0.5 && xa<=0.5 && ya>=-0.5 && ya<=0.5) || (xb>=-0.5 && xb<=0.5 &&
yb>=-0.5 && yb<=0.5) || (xc>=-0.5 && xc<=0.5 && yc>=-0.5 && yc<=0.5) || (xd>=-0.5 &&
xd<=0.5 && yd>=-0.5 && yd<=0.5))
        {
            printf("Intersect with ");
        }

        //up
        if(ya>0.5 || yb>0.5 || yc>0.5 || yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 || yb<-0.5 || yc<-0.5 || yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 || xb>0.5 || xc>0.5 || xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 || xb<-0.5 || xc<-0.5 || xd<-0.5)
        {
            printf("Left ");
        }

        printf("\n");
        break;

    case 'a':
        //glTranslatef(-0.01, 0, 0);
        xa = xa - 0.01;
        xb = xb - 0.01;
        xc = xc - 0.01;
        xd = xd - 0.01;
        glutPostRedisplay();
        //inside
```

```
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }
        else if((xa>=-0.5 && xa<=0.5 && ya>=-0.5 && ya<=0.5) || (xb>=-0.5 && xb<=0.5 &&
yb>=-0.5 && yb<=0.5) || (xc>=-0.5 && xc<=0.5 && yc>=-0.5 && yc<=0.5) || (xd>=-0.5 &&
xd<=0.5 && yd>=-0.5 && yd<=0.5))
        {
            printf("Intersect with ");
        }

        //up
        if(ya>0.5 || yb>0.5 || yc>0.5 || yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 || yb<-0.5 || yc<-0.5 || yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 || xb>0.5 || xc>0.5 || xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 || xb<-0.5 || xc<-0.5 || xd<-0.5)
        {
            printf("Left ");
        }

        printf("\n");
        break;

    case 'd':
        //glTranslatef(0.01, 0, 0);
        xa = xa + 0.01;
        xb = xb + 0.01;
```

```
        xc = xc + 0.01;
        xd = xd + 0.01;
        glutPostRedisplay();
        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }
        else if((xa>=-0.5 && xa<=0.5 && ya>=-0.5 && ya<=0.5) || (xb>=-0.5 && xb<=0.5 &&
yb>=-0.5 && yb<=0.5) || (xc>=-0.5 && xc<=0.5 && yc>=-0.5 && yc<=0.5) || (xd>=-0.5 &&
xd<=0.5 && yd>=-0.5 && yd<=0.5))
        {
           printf("Intersect with ");
        }

        //up
        if(ya>0.5 || yb>0.5 || yc>0.5 || yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 || yb<-0.5 || yc<-0.5 || yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 || xb>0.5 || xc>0.5 || xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 || xb<-0.5 || xc<-0.5 || xd<-0.5)
        {
            printf("Left ");
        }

        printf("\n");
        break;
```

```
   case 'r':
      //glRotatef(1, 0, 0, 1);
      xa = xa*cos(pi/180)-ya*sin(pi/180);
      ya = xa*sin(pi/180)+ya*cos(pi/180);
      xb = xb*cos(pi/180)-yb*sin(pi/180);
      yb = xb*sin(pi/180)+yb*cos(pi/180);
      xc = xc*cos(pi/180)-yc*sin(pi/180);
      yc = xc*sin(pi/180)+yc*cos(pi/180);
      xd = xd*cos(pi/180)-yd*sin(pi/180);
      yd = xd*sin(pi/180)+yd*cos(pi/180);
      glutPostRedisplay();
      //inside
      if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
      {
         printf("Inside ");
      }

      //up
      if(ya>0.5 && yb>0.5 && yc>0.5 && yd>0.5)
      {
         printf("Up ");
      }

      //down
      if(ya<-0.5 && yb<-0.5 && yc<-0.5 && yd<-0.5)
      {
         printf("Down ");
      }

      //right
      if(xa>0.5 && xb>0.5 && xc>0.5 && xd>0.5)
      {
         printf("Right ");
      }

      //down
      if(xa<-0.5 && xb<-0.5 && xc<-0.5 && xd<-0.5)
      {
         printf("Left ");
      }

      printf("\n");
```

```
        break;

    case 'c':
        //glRotatef(-1, 0, 0, 1);
        xa = xa*cos(-pi/180)-ya*sin(-pi/180);
        ya = xa*sin(-pi/180)+ya*cos(-pi/180);
        xb = xb*cos(-pi/180)-yb*sin(-pi/180);
        yb = xb*sin(-pi/180)+yb*cos(-pi/180);
        xc = xc*cos(-pi/180)-yc*sin(-pi/180);
        yc = xc*sin(-pi/180)+yc*cos(-pi/180);
        xd = xd*cos(-pi/180)-yd*sin(-pi/180);
        yd = xd*sin(-pi/180)+yd*cos(-pi/180);
        glutPostRedisplay();
        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }

        //up
        if(ya>0.5 && yb>0.5 && yc>0.5 && yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 && yb<-0.5 && yc<-0.5 && yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 && xb>0.5 && xc>0.5 && xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 && xb<-0.5 && xc<-0.5 && xd<-0.5)
        {
            printf("Left ");
        }
```

```
        printf("\n");
        break;

    case 'm':
        //glScalef(1.1, 1.1, 1);
        xa = xa*1.1;
        ya = ya*1.1;
        xb = xb*1.1;
        yb = yb*1.1;
        xc = xc*1.1;
        yc = yc*1.1;
        xd = xd*1.1;
        yd = yd*1.1;
        glutPostRedisplay();
        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }

        //up
        if(ya>0.5 && yb>0.5 && yc>0.5 && yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 && yb<-0.5 && yc<-0.5 && yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 && xb>0.5 && xc>0.5 && xd>0.5)
        {
            printf("Right ");
        }

        //down
        if(xa<-0.5 && xb<-0.5 && xc<-0.5 && xd<-0.5)
        {
```

```
            printf("Left ");
        }


        printf("\n");
        break;

    case 'n':
        //glScalef(0.9, 0.9, 1);
        xa = xa*0.9;
        ya = ya*0.9;
        xb = xb*0.9;
        yb = yb*0.9;
        xc = xc*0.9;
        yc = yc*0.9;
        xd = xd*0.9;
        yd = yd*0.9;
        glutPostRedisplay();
        //inside
        if(xa>=-0.5 && xa<=0.5 && xb>=-0.5 && xb<=0.5 && xc>=-0.5 && xc<=0.5 && xd>=-0.5 &&
xd<=0.5 && ya>=-0.5 && ya<=0.5 && yb>=-0.5 && yb<=0.5 && yc>=-0.5 && yc<=0.5 &&
yd>=-0.5 && yd<=0.5)
        {
            printf("Inside ");
        }

        //up
        if(ya>0.5 && yb>0.5 && yc>0.5 && yd>0.5)
        {
            printf("Up ");
        }

        //down
        if(ya<-0.5 && yb<-0.5 && yc<-0.5 && yd<-0.5)
        {
            printf("Down ");
        }

        //right
        if(xa>0.5 && xb>0.5 && xc>0.5 && xd>0.5)
        {
            printf("Right ");
        }
```

```
        //down
        if(xa<-0.5 && xb<-0.5 && xc<-0.5 && xd<-0.5)
        {
            printf("Left ");
        }

        printf("\n");
        break;
    }
}

void special(int key, int x, int y)
{
    switch(key)
    {
    case GLUT_KEY_UP:
        glTranslatef(0, 0.01, 0);
        glutPostRedisplay();
        break;

    case GLUT_KEY_DOWN:
        glTranslatef(0, -0.01, 0);
        glutPostRedisplay();
        break;

    case GLUT_KEY_LEFT:
        glTranslatef(-0.01, 0, 0);
        glutPostRedisplay();
        break;

    case GLUT_KEY_RIGHT:
        glTranslatef(0.01, 0, 0);
        glutPostRedisplay();
        break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("CSE 4-2 B1");
```

```
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(special);
    glutMainLoop();
    return 0;
}
```

Feb 20

```
#include <stdio.h>

int main(void)
{
   float x1, y1, x2, y2;
   float minx, maxx;
   scanf("%f %f %f %f",&x1, &y1, &x2, &y2);

   float m = (y2 - y1) / (x2 - x1);
   float c = y1 - m * x1;

   if(x1 > x2)
   {
      minx = x2;
      maxx = x1;
   }
```

```
    else
    {
        minx = x1;
        maxx = x2;
    }

    for (float x = minx; x<= maxx; x++)
    {
        float y = round(m*x + c);
        printf("%.0f %.0f\n", x, y);
    }
    return 0;
}
```

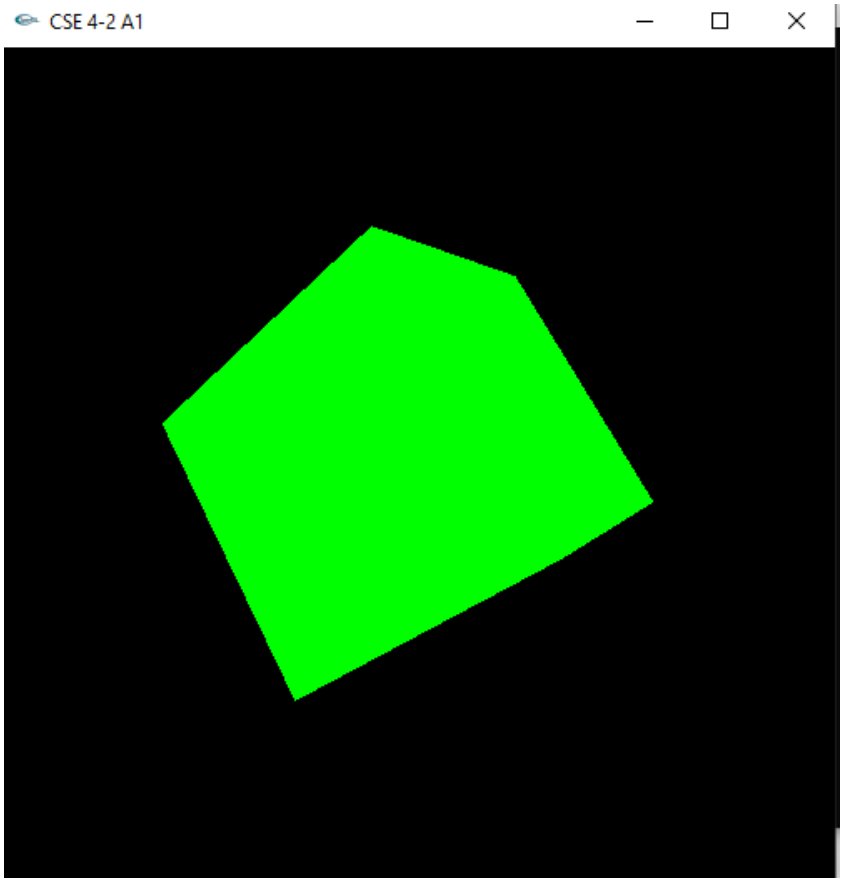
Feb 20

```
#include <stdio.h>
#include <math.h>
#define pi 3.14159265

int main(void)
{
    float r, cx, cy;
    scanf("%f %f %f", &r, &cx, &cy);

    for (float theta = 0; theta < 360; theta = theta + 5)
    {
        float x = round(r*cos((pi * theta / 180)) + cx);
        float y = round(r*sin((pi * theta / 180)) + cy);
        printf("%.0f %.0f\n", x, y);
    }

    return 0;
}
```

Feb 27

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

float angle = 0;

void spin(void)
{
    angle = angle + 0.1;
    glutPostRedisplay();
}
```

```
void display(void)
{

   glClear(GL_COLOR_BUFFER_BIT);
   glLoadIdentity();
   gluLookAt(0, 0, 5, 0, 0, 0, 0, 1, 0);
   glRotatef(angle, 1, 0, 0);
   glRotatef(angle, 0, 1, 0);
   glRotatef(angle, 0, 0, 1);
   glColor3f(0, 1, 0);
   glutSolidCube(2.0);
   glFlush();
}

void reshape(int w, int h)
{
   glViewport(0, 0, w, h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluPerspective(60, w/h, 1, 100);
   glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowSize(500, 500);
   glutInitWindowPosition(100,100);
   glutCreateWindow("CSE 4-2 A1");
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutIdleFunc(spin);
   glutMainLoop();
   return 0;
}
```



Mar 6

```
#include <stdio.h>

int main(void)
{
   float xmax, xmin, ymax, ymin;
   float x1, y1, x2, y2;
   int flag1, flag2;
   scanf("%f%f%f%f",&xmax, &xmin, &ymax, &ymin);
   scanf("%f%f%f%f", &x1, &y1, &x2, &y2);

   //First End Point
   if(x1>xmin && x1<xmax && y1>ymin && y1<ymax)
   {
      flag1 = 0; //inside
   }
   else if(x1<=xmin && y1>=ymax)
   {
      flag1 = 5; //up-left
   }
   else if(x1>=xmax && y1>=ymax)
```

```
{
   flag1 = 6; //up-right
}
else if(x1<=xmin && y1<=ymin)
{
   flag1 = 7; //down-left
}
else if(x1>=xmax && y1<=ymin)
{
   flag1 = 8; //down-right
}
else if(y1>=ymax)
{
   flag1 = 1; //up
}
else if(y1<=ymin)
{
   flag1 = 2; //down
}
else if(x1>=xmax)
{
   flag1 = 4; //right
}
else if(x1<=xmin)
{
   flag1 = 3; //left
}


//Second End Point

if(x2>xmin && x2<xmax && y2>ymin && y2<ymax)
{
   flag2 = 0; //inside
}
else if(x2<=xmin && y2>=ymax)
{
   flag2 = 5; //up-left
}
else if(x2>=xmax && y2>=ymax)
{
   flag2 = 6; //up-right
}
else if(x2<=xmin && y2<=ymin)
{
   flag2 = 7; //down-left
}
```

```
    else if(x2>=xmax && y2<=ymin)
    {
        flag2 = 8; //down-right
    }
    else if(y2>=ymax)
    {
        flag2 = 1; //up
    }
    else if(y2<=ymin)
    {
        flag2 = 2; //down
    }
    else if(x2>=xmax)
    {
        flag2 = 4; //right
    }
    else if(x2<=xmin)
    {
        flag2 = 3; //left
    }

    printf("\n%d %d",flag1, flag2);
    return 0;
}
```

Apr 3

```
#include <stdio.h>

int main(void)
{
    float xmax, xmin, ymax, ymin;
    float x1, y1, x2, y2, x3, y3, x4, y4;
    int flag1, flag2;
    scanf("%f%f%f%f",&xmax, &xmin, &ymax, &ymin);
    scanf("%f%f%f%f", &x1, &y1, &x2, &y2);

    //First End Point
    if(x1>xmin && x1<xmax && y1>ymin && y1<ymax)
    {
        flag1 = 0; //inside
        printf("%f %f", x1, y1);
    }
    else if(x1<=xmin && y1>=ymax)
    {
        flag1 = 5; //up-left
```

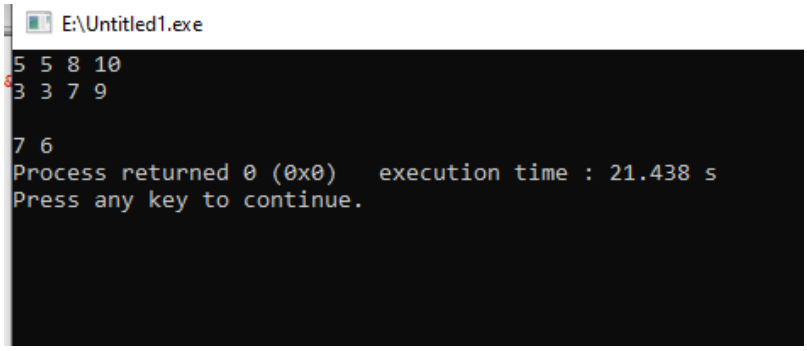
```
    x3 = xmin;
    y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));

    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

    x4 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y4 = ymax;

    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x1>=xmax && y1>=ymax)
{
    flag1 = 6; //up-right

    x3 = xmax;
    y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

    x4 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y4 = ymax;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x1<=xmin && y1<=ymin)
{
    flag1 = 7; //down-left
    x3 = xmin;
    y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

    x4 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
    y4 = ymin;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
```

```
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x1>=xmax && y1<=ymin)
{
    flag1 = 8; //down-right
    x3 = xmax;
    y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

    x4 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
    y4 = ymin;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(y1>=ymax)
{
    flag1 = 1; //up
    x3 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y3 = ymax;
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }
}
else if(y1<=ymin)
{
    flag1 = 2; //down
    x3 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
    y3 = ymin;
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

}
else if(x1>=xmax)
{
    flag1 = 4; //right
    x3 = xmax;
    y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
```

```
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }
}
else if(x1<=xmin)
{
    flag1 = 3; //left
    x3 = xmin;
    y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }
}


//Second End Point

if(x2>xmin && x2<xmax && y2>ymin && y2<ymax)
{
    flag2 = 0; //inside
    printf("%f %f\n", x2, y2);
}
else if(x2<=xmin && y2>=ymax)
{
    flag2 = 5; //up-left
    x3 = xmin;
    y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }

    x4 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y4 = ymax;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x2>=xmax && y2>=ymax)
{
    flag2 = 6; //up-right
    x3 = xmax;
    y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
```

```
    {
        printf("%f %f\n",x3, y3);
    }
    x4 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y4 = ymax;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x2<=xmin && y2<=ymin)
{
    flag2 = 7; //down-left
    x3 = xmin;
    y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }
    x4 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
    y4 = ymin;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(x2>=xmax && y2<=ymin)
{
    flag2 = 8; //down-right
    x3 = xmax;
    y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
    if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
    {
        printf("%f %f\n",x3, y3);
    }
    x4 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
    y4 = ymin;
    if(x4>=xmin && x4<=xmax && y4>=ymin && y4<=ymax)
    {
        printf("%f %f\n",x4, y4);
    }
}
else if(y2>=ymax)
{
    flag2 = 1; //up
    x3 = x1+(x2-x1)*((ymax-y1)/(y2-y1));
    y3 = ymax;
```

```
        if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
        {
            printf("%f %f\n",x3, y3);
        }
    }
    else if(y2<=ymin)
    {
        flag2 = 2; //down
        x3 = x1+(x2-x1)*((ymin-y1)/(y2-y1));
        y3 = ymin;
        if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
        {
            printf("%f %f\n",x3, y3);
        }
    }
    else if(x2>=xmax)
    {
        flag2 = 4; //right
        x3 = xmax;
        y3 = y1+(y2-y1)*((xmax-x1)/(x2-x1));
        if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
        {
            printf("%f %f\n",x3, y3);
        }
    }
    else if(x2<=xmin)
    {
        flag2 = 3; //left
         x3 = xmin;
        y3 = y1+(y2-y1)*((xmin-x1)/(x2-x1));
        if(x3>=xmin && x3<=xmax && y3>=ymin && y3<=ymax)
        {
            printf("%f %f\n",x3, y3);
        }
    }

    printf("\n%d %d",flag1, flag2);
    return 0;
}
```

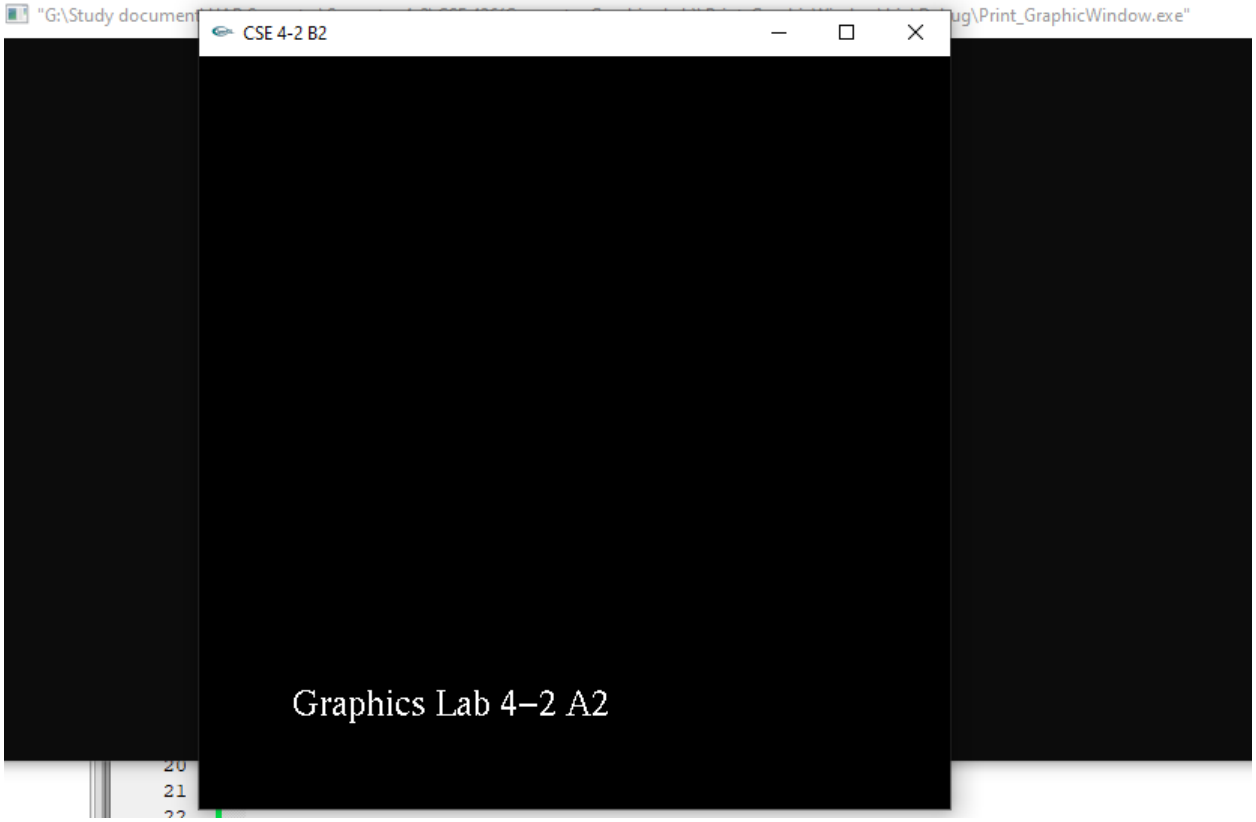


Apr 10

Print in Graphic Window

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

void display(void)
{

   glFlush();
}


void drawString(void *font, float x, float y, char *str)
{

   char *ch;
   glRasterPos3f(x, y,0);
   for(ch=str; *ch; ch++)
   {
      glutBitmapCharacter(font, (int)*ch);
   }

}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowSize(500, 500);
   glutInitWindowPosition(100,100);
   glutCreateWindow("CSE 4-2 B2");
   drawString(GLUT_BITMAP_TIMES_ROMAN_24, -0.75, -0.75, "Graphics Lab 4-2 A2");
   glutDisplayFunc(display);

   glutMainLoop();
```

```
    return 0;
}
```


"G:\Study document
ug\Print_GraphicWindow.exe"
CSE 4-2 B2
Graphics Lab 4–2 A2
20
21