Quiz #3 (CSE4190.410)

October 7, 2013 (Monday)

Name: _____ **Dept:** _____ **ID No:** _____

1. (10 points) Fill in the blanks in the following OpenGL program.

```
#include <GL/glut.h>
#include "curve.h"

CubicBezierCurve curve;
GLsizei width = 640, height = 480;
int edit_ctrlpts_idx = -1;

int hit_index(CubicBezierCurve *curve, int x, int y)
{
   for (int i = 0; i < 4; i++)
   {
      REAL tx = curve->control_pts[i][0] - x;
      REAL ty = curve->control_pts[i][1] - y;
      if (_____________________________) return i;
   }
   return -1;
}

void reshape_callback(GLint nw, GLint nh)
{
   width = nw;
   height = nh;
   glViewport(0, 0, width, height);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0, width, 0, height);
}

void mouse_callback(GLint button, GLint action, GLint x, GLint y)
{
   if (GLUT_LEFT_BUTTON == button)
   {
      switch (action)
      {
         case GLUT_DOWN:
            edit_ctrlpts_idx = _________________(&curve, x, height - y);
            break
         case GLUT_UP:
            edit_ctrlpts_idx = -1;
            break
         default: break
      }
   }
   glutPostRedisplay();
}
```

```
void mouse_move_callback(GLint x, GLint y)
{
   if (edit_ctrlpts_idx != -1)
   {
      curve.control_pts[edit_ctrlpts_idx][0] = _______________;
      curve.control_pts[edit_ctrlpts_idx][1] = _______________;
   }
   glutPostRedisplay();
}


void display_callback()
{
#define RES 100

   /* curve */
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3ub(0, 0, 0);
   glBegin(____________________);
   for (int i = 0; i <= RES; i++)
   {
   Point pt;
      const REAL t = (REAL)i / (REAL)RES;
      evaluate(&curve, t, pt);
      glVertex2f(pt[0], pt[1]);
   }
   glEnd();

   /* control mesh */
   glColor3ub(_____,_____,_____);
   glBegin(____________________);
   for (int i = 0; i < 4; i++)
   {
      REAL *pt = curve.control_pts[i];
      glVertex2f(pt[0], pt[1]);
   }
   glEnd();

   /* control pts */
   glColor3ub(_____,_____,_____);
   glPointSize(10.0);
   glBegin(____________________);
   for (int i = 0; i < 4; i++)
   {
      REAL *pt = curve.control_pts[i];
      glVertex2f(pt[0], pt[1]);
   }
   glEnd();
   glut_______________();
}
```

```
void keyboard_callback(unsigned char key, int x, int y)
{
   switch(key)
   {
   case (27): exit(0); break
   default: break
   }
   glutPostRedisplay();
}

int main(int argc, char *argv[])
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
   glutInitWindowSize(width, height);
   glutCreateWindow("Beizer Editor");
   init();
   glutReshapeFunc(reshape_callback);
   glutMouseFunc(mouse_callback);
   glutMotionFunc(mouse_move_callback);
   glutDisplayFunc(display_callback);
   glutKeyboardFunc(keyboard_callback);
   glutMainLoop();
   return 0;
}
```