R.V. COLLEGE OF ENGINEERING, BENGALURU
(Autonomous Institution Affiliated to VTU, Belagavi)

COMPUTER GRAPHICS ASSIGNMENT REPORT ON

SELECTION SORT SIMULATION

Submitted by

Jeevan D C **1RV13CS062**

Under the guidance of

Dr. Krishnappa H. K. , Associate Professor, CSE

Submitted to

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
R.V. College of Engineering, Bengaluru-560059

Program for graphical simulation of Selection Sort.

```
#include<iostream>
#include<string>
using namespace std;
#include<GL/glut.h>
int LEN;
int arr[100];

void delay(){
	int c = 1, d = 1, k;

	for (c = 1; c <= 32767; c++)
	for (d = 1; d <= 32767; d++)
	{
	}

}

string buildstr(){
	string st = "";
	for (int i = 0; i < LEN; i++){
		st += std::to_string(arr[i]);
		st += " ";
	}
	return st;
}


void drawBitmapText(string str, float x, float y, float z)
{
	string c;
	glRasterPos3f(x, y, z);
```

```
	for (int i = 0; i < str.length(); i++)
	{
		glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_10, str[i]);
	}
}

void drawBitmapTextMsg(string str, float x, float y, float z)
{
	string c;
	glRasterPos3f(x, y, z);

	for (int i = 0; i < str.length(); i++)
	{
		glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, str[i]);
	}
}

void drawBitmapNum(string str, float x, float y, float z)
{
	string c;
	glRasterPos3f(x, y, z);

	for (int i = 0; i < str.length(); i++)
	{
		glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, str[i]);
	}
}


void drawStrokeText(char*string, int x, int y, int z)
{
	char *c;
	glPushMatrix();
	glTranslatef(x, y + 8, z);
```

```
	glScalef(0.09f, -0.08f, z);

	for (c = string; *c != '\0'; c++)
	{
		glutStrokeCharacter(GLUT_STROKE_ROMAN, *c);
	}
	glPopMatrix();
}

void drawpolygon(){
	glColor3f(1, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(20, 20);
	glVertex2i(40, 30);
	glVertex2i(40, 20);
	glVertex2i(20, 30);
	glEnd();
	glColor3f(.5, .5, .5);
	drawBitmapText("CURRENT INDEX", 50, 30, 0);

	glColor3f(0, 0, 1);
	glBegin(GL_POLYGON);
	glVertex2i(170, 20);
	glVertex2i(190, 30);
	glVertex2i(190, 20);
	glVertex2i(170, 30);
	glEnd();
	glColor3f(.5, .5, .5);
	drawBitmapText("SWAP INDEX", 200, 30, 0);

}

void sort(){
	glColor3f(0, 1, 0);
```

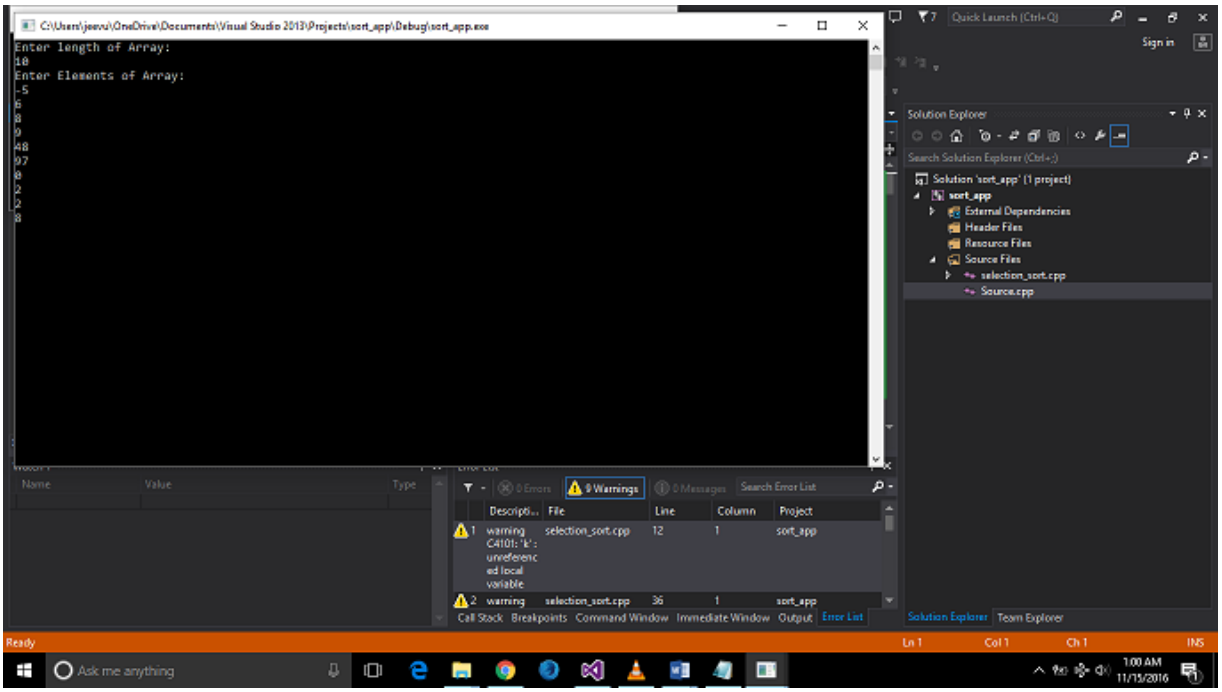
```
string st = "";

int c, d, swap, position, n = LEN;
for (c = 0; c < (n - 1); c++)
{
	position = c;

	if (c == n - 2)
	{
		glColor3f(.7, .4, .5);
		drawBitmapTextMsg("Finishing Sorting!!", 250, 180, 0);
		delay(); delay(); delay();

	}



	for (d = c + 1; d < n; d++)
	{
		if (arr[position] > arr[d])
			position = d;
	}

	for (int i = 0; i < LEN; i++){

		drawpolygon();
		if (c == i) glColor3f(1, 0, 0);
		else glColor3f(0, 1, 0);

		drawBitmapNum(to_string(arr[i]), 125 + i * 40, 125, 0);
		if (position == i) glColor3f(0, 0, 1);

		drawBitmapNum(to_string(arr[i]), 125 + i * 40, 125, 0);
	}
	glutSwapBuffers();
```

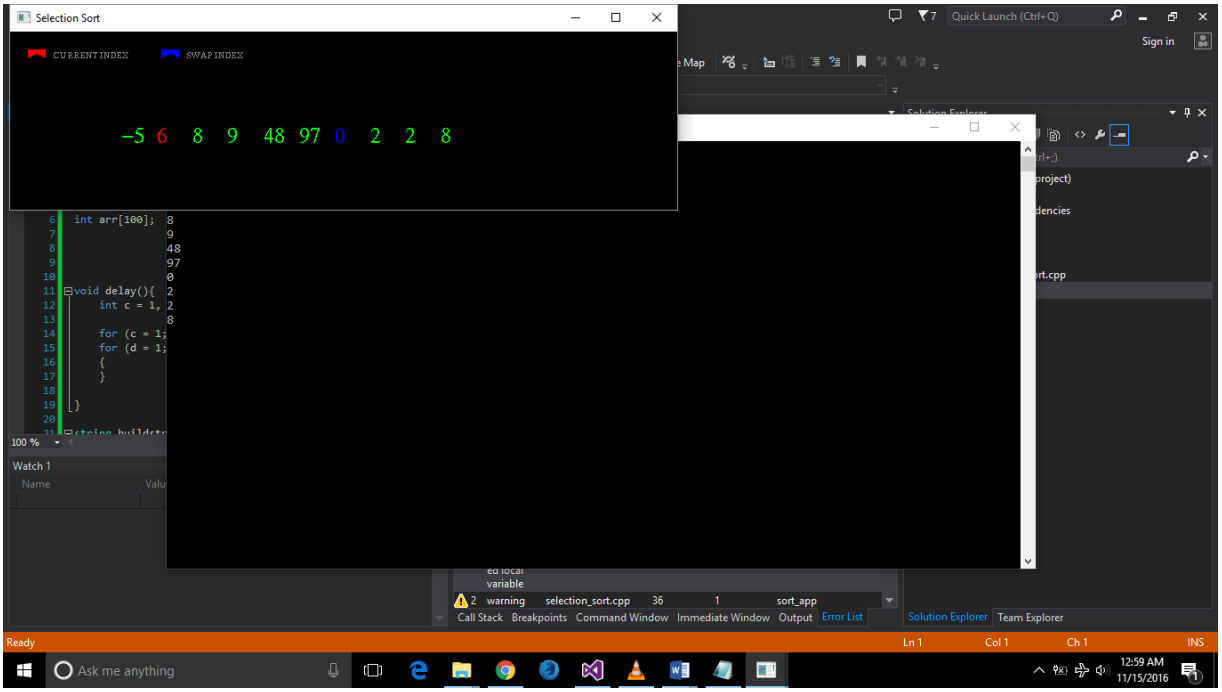
```
                glutPostRedisplay();
                glClear(GL_COLOR_BUFFER_BIT);
                delay();
                delay();

                if (position != c)
                {
                        swap = arr[c];
                        arr[c] = arr[position];
                        arr[position] = swap;
                }



        }


}

void init()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glClearColor(0, 0, 0, 0);
        gluOrtho2D(0, 750, 0, 200);
}

void reshape(int w, int h)
{

        glViewport(0, 0, w, h);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0, w, h, 0);
        glMatrixMode(GL_MODELVIEW);
```

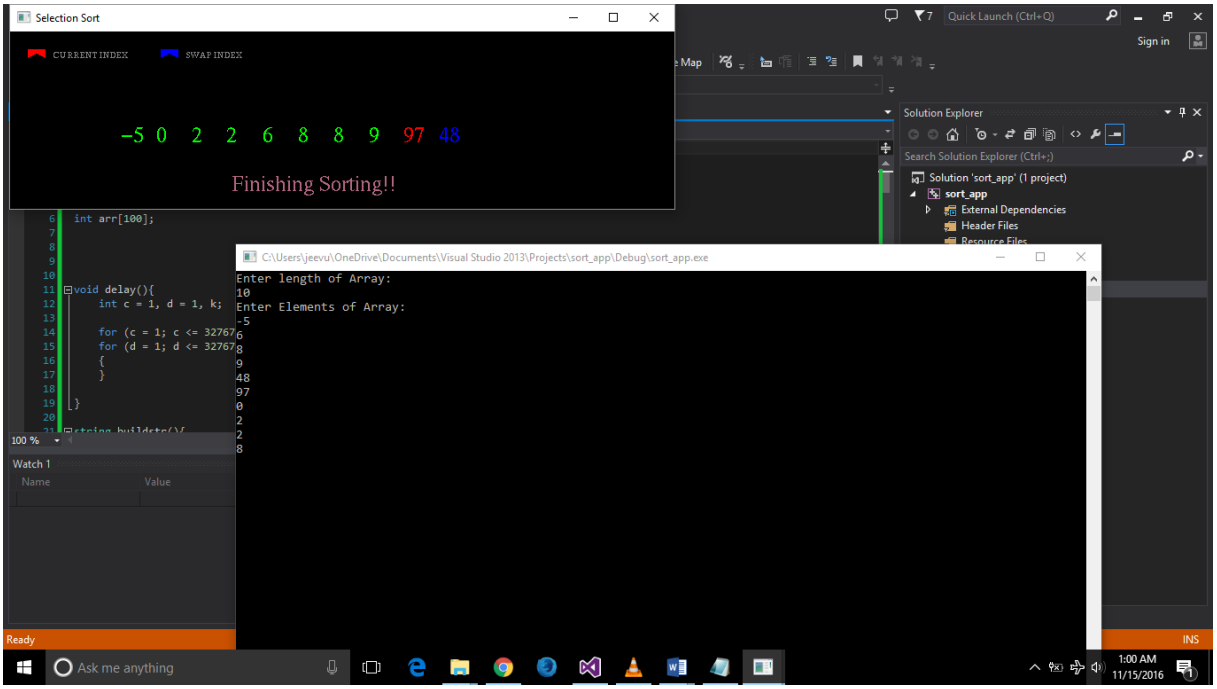
```
	glLoadIdentity();

}

void render(void)
{
	glClear(GL_COLOR_BUFFER_BIT);
	glLoadIdentity();
	sort();
	glutSwapBuffers();
}

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


int main(int argc, char* argv[])
{
	cout << "Enter length of Array:" << endl;
	cin >> LEN;
	cout << "Enter Elements of Array:" << endl;
	for (int i = 0; i < LEN; i++)
		cin >> arr[i];
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
	glutInitWindowSize(750, 200);
	glutInitWindowPosition(0, 0);
	glutCreateWindow("Selection Sort");
	glutDisplayFunc(render);
	glutReshapeFunc(reshape);
```

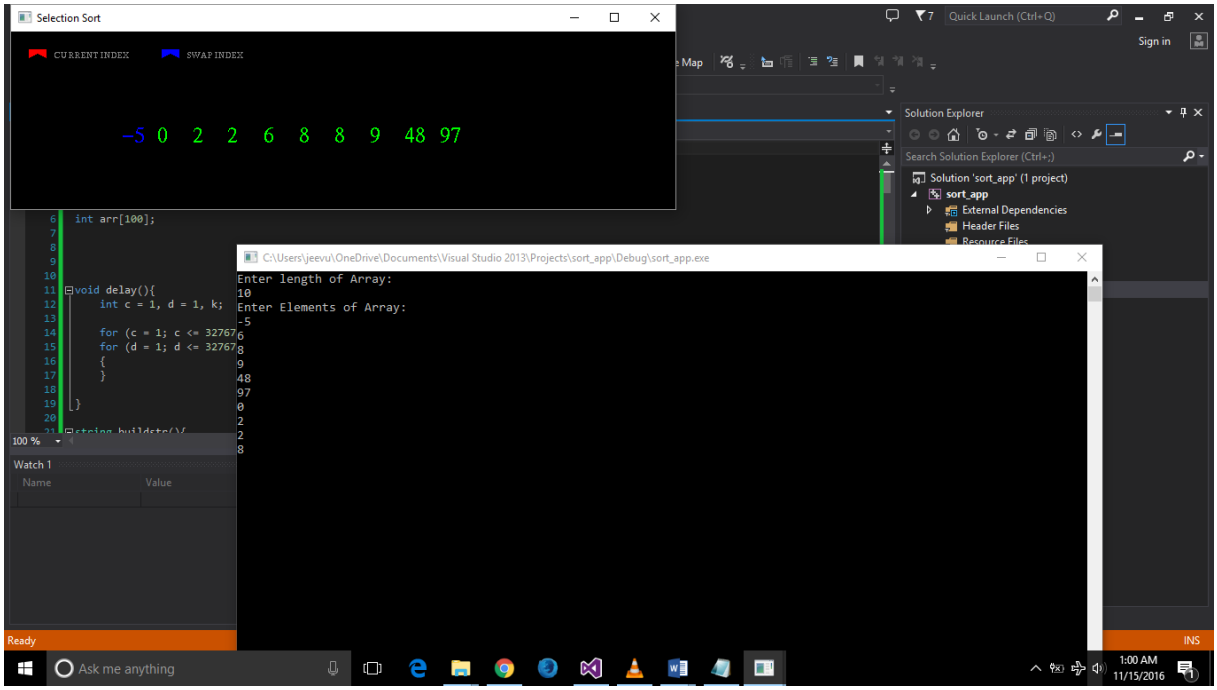
```
    glutKeyboardFunc(keyboard);
    init();
    glutMainLoop();
    return 0;
}
```

Screenshots






Selection Sort
CURRENT INDEX
SWAP INDEX
-5 0 2 2 6 8 8 9 97 48
Finishing Sorting!!
int arr[100];
void delay(){
int c = 1, d = 1, k;
for (c = 1; c <= 32767
for (d = 1; d <= 32767
{
}
}
C:\Users\jeevu\OneDrive\Documents\Visual Studio 2013\Projects\sort_app\Debug\sort_app.exe
Enter length of Array:
10
Enter Elements of Array:
-5
6
8
9
48
97
0
2
2
8
Solution Explorer
Solution 'sort_app' (1 project)
sort_app
External Dependencies
Header Files
Watch 1
Name
Value
Ready
Ask me anything
1:00 AM
11/15/2016


Selection Sort
CURRENT INDEX
SWAP INDEX
-5 0 2 2 6 8 8 9 48 97
int arr[100];
void delay(){
int c = 1, d = 1, k;
for (c = 1; c <= 32767
for (d = 1; d <= 32767
{
}
}
C:\Users\jeevu\OneDrive\Documents\Visual Studio 2013\Projects\sort_app\Debug\sort_app.exe
Enter length of Array:
10
Enter Elements of Array:
-5
6
8
9
48
97
0
2
2
8
Solution Explorer
Solution 'sort_app' (1 project)
sort_app
External Dependencies
Header Files
Watch 1
Name
Value
Ready
Ask me anything
1:00 AM
11/15/2016