## Compiling individual files

To compile individual file use the command:

```
g++ Generate_graph.cpp -o graph
g++ Fibonnaci_heap.cpp -o fibonacci
g++ Normal_array.cpp -o array
g++ STL_heap.cpp -o stl
```

## Generate Graph

To generate-graph (text file) for input having n nodes, d density, s source node.

```
./graph n d s
```

This generates a text file with the name *user_input_s_d.txt*

## To run the Dijkstra algorithm on the input graph

To calculate with a particular algorithm we use after compilation.

```
./fibonacci user_input_s_d.txt
./array user_input_s_d.txt
./stl user_input_s_d.txt
```

## To run all algorithms at once using Makefile

For example, if nodes are 300 density is 80, and the source node is 0.

```
make nodes=300 density=80 source=0
```

Three files will be generated which contains the shortest distance from source nodes.

Namely,

**output_fibonacci.txt**
**output_array.txt**
**output_stl.txt**

To clean the folder use: **make clean**

Sample output:

```
g++ Fibonnaci_heap.cpp -o fibonacci
g++ Normal_array.cpp -o array
g++ STL_heap.cpp -o stl
./fibonacci user_input*
Total execution time using Fibonacci Heap : 0.045627 secs
./array user_input*
Total execution time using Normal array : 0.041107 secs
./stl user_input*
Total execution time using STL set : 0.035998 secs
```