[Data Structure Homework #8]

Submit C program source file, execution screen capture file.

You should check your program at https://repl.it/languages/c

아래 1, 2, 3, 4 는 모두 코드가 나와야 함.

You should submit the codes corresponding to the following problems 1, 2, 3, 4

1. Graph ADT를 (1) adjacency matrix (Ad_Matrix.h) 와 (2) adjacency list (Ad_List.h) 를 이용하여 구현하시오. 다만, adjacency matrix는 vertex의 삭제는 구현하지않아도 됨.

   이 구현한 부분을 이후 문제에서는 활용할 것. 필요한 추가 파일은 알아서 작성하세요. 어떻게 실행하면 되는지 readme.txt 도 같이 제출하세요.

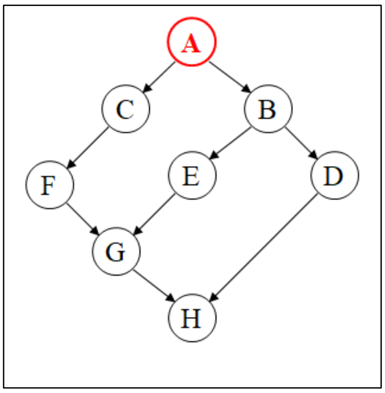
   Implement the Graphs using (1) adjacency matrix (Ad_Matrix.h) and (2) adjacency list (Ad_List.h). You need not implement vertex deletion at adjacency matrix. These graphs should be utilized at the following problems. If you implement additional files, please submit everything. And, please let me know how to execute your program (readme.txt).

2. 다음과 같은 directed 그래프 데이터를 파일로 입력받아, 1의 (1) (2)의 Graph 에 저장한 후, Depth-first traversal로 출력하시오. 또한, Breath-first traversal 로도 출력하시오.

   Get a file with a directed graph data, store it to the graphs implemented in 1 (1) and 1 (2). Print out the graphs by Depthfirst traversal and by Breath-first traversal.

Input File Example: // The starting point : A

입력예: // 아래의 경우 시작은 A



A

A C

A B

B E

B D

C F

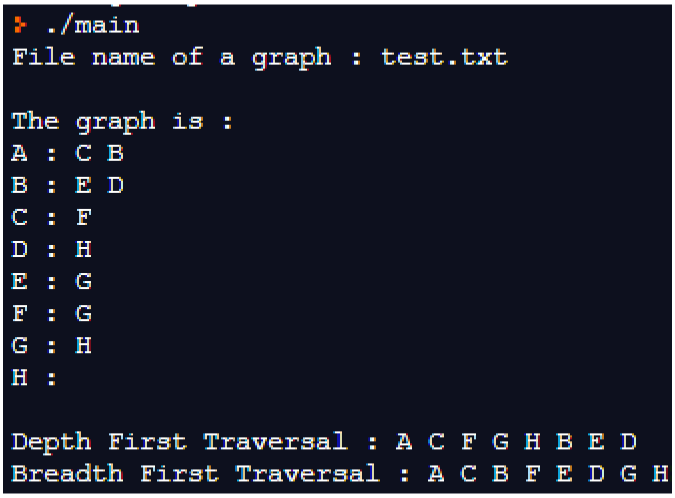
E G

F G

G H

D H

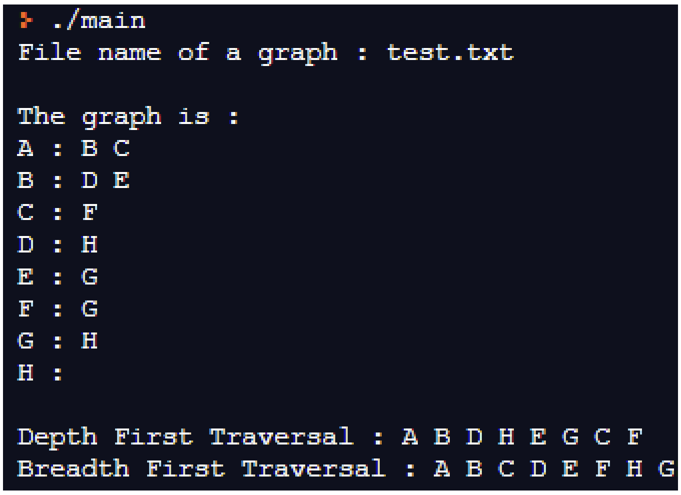
Execution Example : 출력예 (다른 패스가 나올 수 있음):



```
❯ ./main
File name of a graph : test.txt

The graph is :
A : C B
B : E D
C : F
D : H
E : G
F : G
G : H
H :

Depth First Traversal : A C F G H B E D
Breadth First Traversal : A C B F E D G H
```



```
❯ ./main
File name of a graph : test.txt

The graph is :
A : B C
B : D E
C : F
D : H
E : G
F : G
G : H
H :

Depth First Traversal : A B D H E G C F
Breadth First Traversal : A B C D E F H G
```

3. 1 (1)의 adjacency matrix에 그래프의 가중치를 추가하여 구현하시오 (Ad_Matrix3.h).

   Modify the graph implemented in 1 (2) so that its adjacency list has weight. (Ad_Matrix3.h)

4. 3을 이용하여, 다음과 같은 undirected 그래프 데이터를 파일로 입력받아, (1) minimum

spanning tree와 (2) Dijkstra's algorithm으로 shortest path를 구하시오. Dijkstra's algorithm은 과정을 출력으로 보여주어야 함.

Using the undirected graph structure implemented in 3, get a file with graph data and print out the shortest path by (1) minimum spanning tree, and (2) Dijkstra's algorithm.

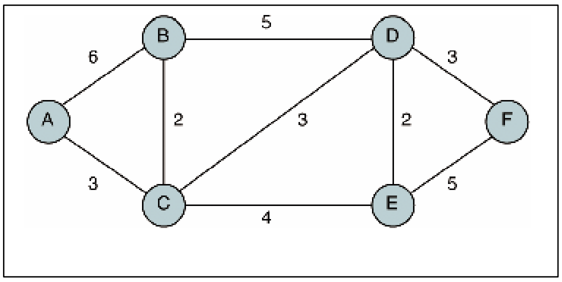
You must show the process of Dijkstra's algorithm.

입력예: // 시작은 A

Input File Example: // The starting point : A

```
A

A B 6
A C 3
B C 2
B D 5
C D 3
C E 4
D E 2
D F 3
E F 5
```

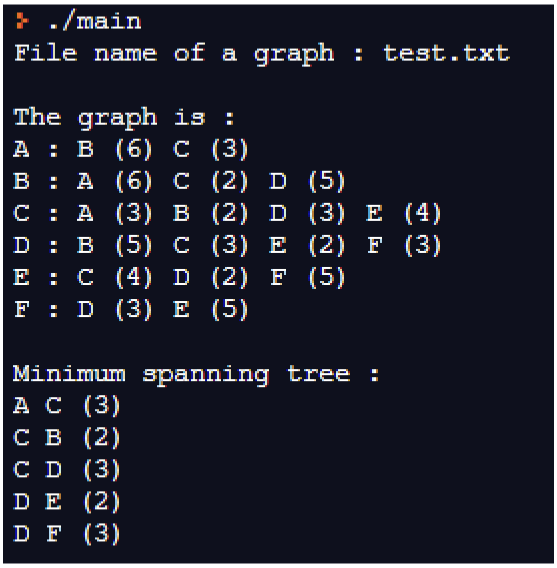


Execution Example:

출력예:

An example of print out:

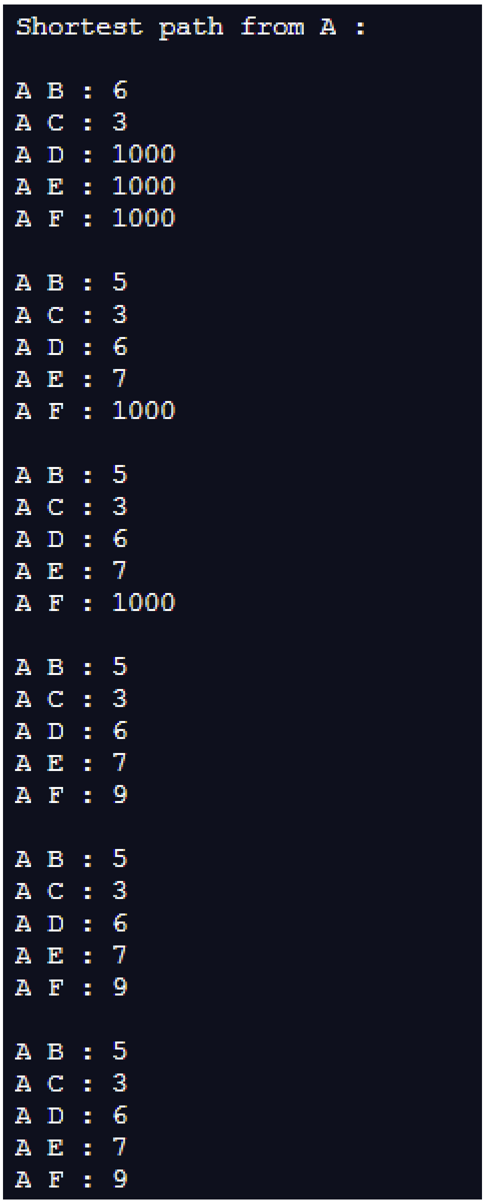
(At this example, the weight is 1000 when there is no direct path between two nodes.)



```
❯ ./main
File name of a graph : test.txt

The graph is :
A : B (6) C (3)
B : A (6) C (2) D (5)
C : A (3) B (2) D (3) E (4)
D : B (5) C (3) E (2) F (3)
E : C (4) D (2) F (5)
F : D (3) E (5)

Minimum spanning tree :
A C (3)
C B (2)
C D (3)
D E (2)
D F (3)
```



```
Shortest path from A :

A B : 6
A C : 3
A D : 1000
A E : 1000
A F : 1000

A B : 5
A C : 3
A D : 6
A E : 7
A F : 1000

A B : 5
A C : 3
A D : 6
A E : 7
A F : 1000

A B : 5
A C : 3
A D : 6
A E : 7
A F : 9

A B : 5
A C : 3
A D : 6
A E : 7
A F : 9

A B : 5
A C : 3
A D : 6
A E : 7
A F : 9
```