**Programming project: shortest path finding**

**• Graph data file :**

We assume that a file with the graph data is given in a text file. The file name is fixed as “graphdata.txt”.

|  |
| --- |
| 12  0 3 8.1 2 2.0 8 2.9  1 4 6.8 7 2.4 2 13.2  2 5 1.3 3 4.9 7 10.8 11 3.2  3 6 5.9 1 5.3 5 3.7  4 10 9.2 11 2.6 2 0.4  5 0 13.1 6 4.2 1 1.5 8 1.1  6 4 3.9 1 7.2  7 4 3.6 9 3.0  8 2 5.1 9 4.1  9 2 1.4 11 7.5  10 7 4.5 9 2.1  11 7 9.1 10 6.0 |

Explanation of the format of the file :

- The first line has the total number of the vertexes of the graph.

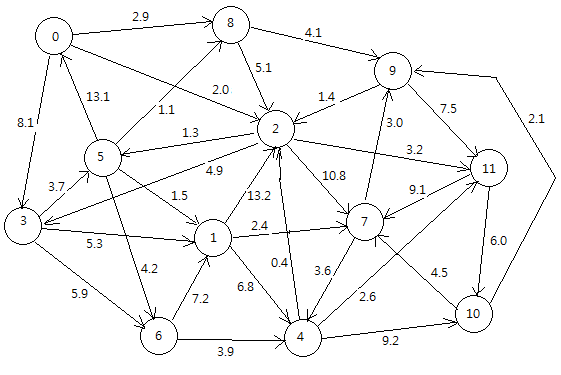
- The lines from the second have the following format :

v u1 c1 u2 c2 ...

Here, v is a vertex number. After that, a list of pairs (ui, ci) follows. There is an arc which comes out of vertex v and enters vertex ui. The cost of this arc is ci. All arcs are directed.

- The number of these lines is the same as the total number of vertexes.

- A example data file given above corresponds to the following graph.



**•** Program development : We need to develop a program that works as follows.

(1) : Read the data file and create the cost matrix COST. Each cell of the matrix has the cost of the corresponding arc. For this task, develop and use the following function.

void Read\_and\_make\_graph ( ) { ....... }

Use the data definition and declaration as follows:

#define Max\_vertex 12

double COST [Max\_vertex][ Max\_vertex] ; // This is declared as a global variable.

(2) : Develop the following function which receives two parameters, start and destination, and finds and prints the shortest path between them. To develop this function, Dijkstra algorithm is modified in such a way that the algorithm stops immediatly when the shortest path to the destination vertex is found.

int shortest\_path (int start, int destination) { ......}

When this function is called, it prints the shortest path from start to destination and also prints the total cost of the path.

Use the variables declared as global variables.

double distance [Max\_vetex ];

int pred [Max\_vetex] ;

int set\_S [Max\_vetex] ; // This represents the set S. An element has 1 if the vertex is in S, and 0 otherwise.

**• An example of execution of the program:**

The main function first calls the function Read\_and\_make\_graph to create the matrix COST. Then the following loop of command is carried out :

Enter two vertexes (start and destination) to find the shortest path> 3 9

Path: 3, 7, 6, 5, 9 Total cost: 47.3

Enter two vertexes (start and destination) to find the shortest path > 6 1

Path: 6, 7, 6, 5, 1 Total cost: 17.3

Enter two vertexes (start and destination) to find the shortest path > 8 3

Path: 8, 7, 6, 5, 3 Total cost: 31.8

....

Enter two vertexes (start and destination) to find the shortest path > -1 -1

Exit the program.