EXECUTION RUNS

```
Graph G(V,E)
V = {0,1,2,3,4}
E = {<0,1,10>,<0,2,3>,<1,3,2>,<1,2,1>,<2,1,4>,<2,3,8>,
     <2,4,2>,<3,4,7>,<4,3,9>,<4,0,4>,<4,1,2>,<1,0,1>}
```

DIJKSTRA'S SINGLE SOURCE SHORTEST PATH ALGORTHM

```
dab@dab-dell-3010:~/Desktop/graph18$ ./dijkstra.out
       Graph Creation [Undirected/Directed]...
              Type of Graph [0: UnDirected] := 1
              How Many Vertices [upto 10 vertices]?? 5

       Vertices starts at 0 and terminates at 4
              Vertex ID of -1 terminates Input

       Enter Existing Edges in the Graph
              -----------------------------------
              Edge#     'u'  'v' Cost Remark
              -----------------------------------
                 1       0    1    10  Edge Taken
                 2       0    2    3   Edge Taken
                 3       1    3    2   Edge Taken
                 4       1    2    1   Edge Taken
                 5       2    1    4   Edge Taken
                 6       2    3    8   Edge Taken
                 7       2    4    2   Edge Taken
                 8       3    4    7   Edge Taken
                 9       4    3    9   Edge Taken
                10       4    0    4   Edge Taken
                11       4    1    2   Edge Taken
                12       1    0    1   Edge Taken
                13      -1   -1    0   Invalid Edge

Graph with  5 vertices ...
       Adjacency Matrix is.... [ 786 denotes INFINITY ] [ -1 denotes no parent ]
                 u|v |   0    1    2    3    4
              -------------------------------
                 0  |   0   10    3  786  786
                 1  |   1    0    1    2  786
                 2  | 786    4    0    8    2
                 3  | 786  786  786    0    7
                 4  |   4    2  786    9    0
```

```
Enter Source Vertex [0 thru 4]: 0
        ---------------------------------------------------
         Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
        ---------------------------------------------------
               DIST[]     |   0  |  10  |   3  | 786  | 786  |
        ---------------------------------------------------
               PREV[]     |  -1  |   0  |   0  |  -1  |  -1  |
        ---------------------------------------------------
               VISITED[]  |   1  |   0  |   0  |   0  |   0  |
        ---------------------------------------------------

        ---------------------------------------------------
         Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
        ---------------------------------------------------
               DIST[]     |   0  |   7  |   3  |  11  |   5  |
        ---------------------------------------------------
               PREV[]     |  -1  |   2  |   0  |   2  |   2  |
        ---------------------------------------------------
               VISITED[]  |   1  |   0  |   1  |   0  |   0  |
        ---------------------------------------------------

        ---------------------------------------------------
         Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
        ---------------------------------------------------
               DIST[]     |   0  |   7  |   3  |  11  |   5  |
        ---------------------------------------------------
               PREV[]     |  -1  |   2  |   0  |   2  |   2  |
        ---------------------------------------------------
               VISITED[]  |   1  |   0  |   1  |   0  |   1  |
        ---------------------------------------------------

        ---------------------------------------------------
         Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
        ---------------------------------------------------
               DIST[]     |   0  |   7  |   3  |   9  |   5  |
        ---------------------------------------------------
               PREV[]     |  -1  |   2  |   0  |   1  |   2  |
        ---------------------------------------------------
               VISITED[]  |   1  |   1  |   1  |   0  |   1  |
        ---------------------------------------------------

        Path: [u]->[v]  Distance  Shortest Path
              --------  --------  -------------
              [0]->[1]       7     0-> 2-> 1
              [0]->[2]       3     0-> 2
              [0]->[3]       9     0-> 2-> 1-> 3
              [0]->[4]       5     0-> 2-> 4

Dijkstra's Algorithm for Different Source?? [0 to Stop] : 1
Enter Source Vertex [0 thru 4]: 4
        ---------------------------------------------------
```

```
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
-------------------------------------------------------
       DIST[]     |   4  |   2  | 786  |   9  |   0  |
-------------------------------------------------------
       PREV[]     |   4  |   4  |  -1  |   4  |  -1  |
-------------------------------------------------------
       VISITED[]  |   0  |   0  |   0  |   0  |   1  |
-------------------------------------------------------

-------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
-------------------------------------------------------
       DIST[]     |   3  |   2  |   3  |   4  |   0  |
-------------------------------------------------------
       PREV[]     |   1  |   4  |   1  |   1  |  -1  |
-------------------------------------------------------
       VISITED[]  |   0  |   1  |   0  |   0  |   1  |
-------------------------------------------------------

-------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
-------------------------------------------------------
       DIST[]     |   3  |   2  |   3  |   4  |   0  |
-------------------------------------------------------
       PREV[]     |   1  |   4  |   1  |   1  |  -1  |
-------------------------------------------------------
       VISITED[]  |   0  |   1  |   1  |   0  |   1  |
-------------------------------------------------------

-------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |
-------------------------------------------------------
       DIST[]     |   3  |   2  |   3  |   4  |   0  |
-------------------------------------------------------
       PREV[]     |   1  |   4  |   1  |   1  |  -1  |
-------------------------------------------------------
       VISITED[]  |   1  |   1  |   1  |   0  |   1  |
-------------------------------------------------------

Path: [u]->[v]  Distance  Shortest Path
      --------  --------  -------------
      [4]->[0]       3     4-> 1-> 0
      [4]->[1]       2     4-> 1
      [4]->[2]       3     4-> 1-> 2
      [4]->[3]       4     4-> 1-> 3

Dijkstra's Algorithm for Different Source?? [0 to Stop] : 0
```

FLOYD-WARSHALL ALL-PAIRS SHORTEST PATH ALGORITHM

```
dab@dab-dell-3010:~/Desktop/graph18$  ./floyd_warshall.out
       Graph Creation [Undirected/Directed]...
              Type of Graph [0: UnDirected] := 1
              How Many Vertices [upto 10 vertices]?? 5

       Vertices starts at 0 and terminates at 4, Vertex ID of -1 terminates Input
       Enter Existing Edges in the Graph
              -----------------------------------
              Edge#     'u'  'v' Cost  Remark
              -----------------------------------
                 1       0    1    10  Edge Taken
                 2       0    2    3   Edge Taken
                 3       1    3    2   Edge Taken
                 4       1    2    1   Edge Taken
                 5       2    1    4   Edge Taken
                 6       2    3    8   Edge Taken
                 7       2    4    2   Edge Taken
                 8       3    4    7   Edge Taken
                 9       4    3    9   Edge Taken
                10       4    0    4   Edge Taken
                11       4    1    2   Edge Taken
                12       1    0    1   Edge Taken
                13      -1   -1    0   Invalid Edge
Graph with  5 vertices ...
       Adjacency Matrix is.... [ 786 denotes INFINITY ]
                 u|v |   0    1    2    3    4
              -------------------------------
                 0  |   0   10    3  786  786
                 1  |   1    0    1    2  786
                 2  | 786    4    0    8    2
                 3  | 786  786  786    0    7
                 4  |   4    2  786    9    0
At Initialization ...
       Cost Matrix is....
                 u|v |   0    1    2    3    4
              -------------------------------
                 0  |   0   10    3  786  786
                 1  |   1    0    1    2  786
                 2  | 786    4    0    8    2
                 3  | 786  786  786    0    7
                 4  |   4    2  786    9    0
       Path Matrix is....
                 u|v |   0    1    2    3    4
              -------------------------------
                 0  |   0    1    2    3    4
                 1  |   0    1    2    3    4
                 2  |   0    1    2    3    4
                 3  |   0    1    2    3    4
                 4  |   0    1    2    3    4
```

```
ITERATION  K = 1
       Cost Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0   10    3  786  786
                  1  |   1    0    1    2  786
                  2  | 786    4    0    8    2
                  3  | 786  786  786    0    7
                  4  |   4    2    7    9    0
       Path Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    1    2    3    4
                  1  |   0    1    2    3    4
                  2  |   0    1    2    3    4
                  3  |   0    1    2    3    4
                  4  |   0    1    0    3    4
ITERATION  K = 2
       Cost Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0   10    3   12  786
                  1  |   1    0    1    2  786
                  2  |   5    4    0    6    2
                  3  | 786  786  786    0    7
                  4  |   3    2    3    4    0
       Path Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    1    2    1    4
                  1  |   0    1    2    3    4
                  2  |   1    1    2    1    4
                  3  |   0    1    2    3    4
                  4  |   1    1    1    1    4
ITERATION  K = 3
       Cost Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    7    3    9    5
                  1  |   1    0    1    2    3
                  2  |   5    4    0    6    2
                  3  | 786  786  786    0    7
                  4  |   3    2    3    4    0
       Path Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    2    2    2    2
                  1  |   0    1    2    3    2
                  2  |   1    1    2    1    4
                  3  |   0    1    2    3    4
                  4  |   1    1    1    1    4
```

```
ITERATION  K = 4
       Cost Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    7    3    9    5
                  1  |   1    0    1    2    3
                  2  |   5    4    0    6    2
                  3  | 786  786  786    0    7
                  4  |   3    2    3    4    0
       Path Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    2    2    2    2
                  1  |   0    1    2    3    2
                  2  |   1    1    2    1    4
                  3  |   0    1    2    3    4
                  4  |   1    1    1    1    4
ITERATION  K = 5
       Cost Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    7    3    9    5
                  1  |   1    0    1    2    3
                  2  |   5    4    0    6    2
                  3  |  10    9   10    0    7
                  4  |   3    2    3    4    0
       Path Matrix is....
                u|v |   0    1    2    3    4
               ------------------------------
                  0  |   0    2    2    2    2
                  1  |   0    1    2    3    2
                  2  |   1    1    2    1    4
                  3  |   4    4    4    3    4
                  4  |   1    1    1    1    4
```

EXECUTION RUNS

```
Graph G(V,E)
V = {0,1,2,3,4,5}
E = {(0,1,6),(0,2,1),(0,3,5),(1,2,5),(1,4,3),
     (2,3,5),(2,4,6),(2,5,4),(3,5,2),(4,5,6)}
```

PRIM'S MINIMUM COST SPANNING TREE

```
dab@dab-dell-3010:~/Desktop/graph18$ ./primsMST.out

Graph Creation [Undirected/Directed]...
        Type of Graph [0: UnDirected] := 0
        How Many Vertices [upto 10 vertices]?? 6

Vertices starts at 0 and terminates at 5, Vertex ID of -1 terminates Input

Enter Existing Edges in the Graph
        ----------------------------------
        Edge#     'u'  'v' Cost Remark
        ----------------------------------
          1       0    1    6      Edge Taken
          2       0    2    1      Edge Taken
          3       0    3    5      Edge Taken
          4       1    2    5      Edge Taken
          5       1    4    3      Edge Taken
          6       2    3    5      Edge Taken
          7       2    4    6      Edge Taken
          8       2    5    4      Edge Taken
          9       3    5    2      Edge Taken
         10       4    5    6      Edge Taken
         11      -1   -1    0      Invalid Edge

Graph with  6 vertices ...
        Adjacency Matrix is.... [ 786 denotes INFINITY ] [ MST[] denotes parent vertex index ]
           u|v |   0    1    2    3    4    5
        -------------------------------------
           0  |   0    6    1    5  786  786
           1  |   6    0    5  786    3  786
           2  |   1    5    0    5    6    4
           3  |   5  786    5    0  786    2
           4  | 786    3    6  786    0    6
           5  | 786  786    4    2    6    0

        ------------------------------------------------------------
         Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |  [5] |
        ------------------------------------------------------------
              DIST[]     |   0  |   6  |   1  |   5  | 786  | 786  |
        ------------------------------------------------------------
               MST[]     |  -1  |   0  |   0  |   0  |  -1  |  -1  |
        ------------------------------------------------------------
              VISITED[]  |   1  |   0  |   0  |   0  |   0  |   0  |
        ------------------------------------------------------------
```

```
-------------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |  [5] |
-------------------------------------------------------------
       DIST[]     |   0  |   5  |   1  |   5  |   6  |   4  |
-------------------------------------------------------------
        MST[]     |  -1  |   2  |   0  |   0  |   2  |   2  |
-------------------------------------------------------------
       VISITED[]  |   1  |   0  |   1  |   0  |   0  |   0  |
-------------------------------------------------------------

-------------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |  [5] |
-------------------------------------------------------------
       DIST[]     |   0  |   5  |   1  |   2  |   6  |   4  |
-------------------------------------------------------------
        MST[]     |  -1  |   2  |   0  |   5  |   2  |   2  |
-------------------------------------------------------------
       VISITED[]  |   1  |   0  |   1  |   0  |   0  |   1  |
-------------------------------------------------------------

-------------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |  [5] |
-------------------------------------------------------------
       DIST[]     |   0  |   5  |   1  |   2  |   6  |   4  |
-------------------------------------------------------------
        MST[]     |  -1  |   2  |   0  |   5  |   2  |   2  |
-------------------------------------------------------------
       VISITED[]  |   1  |   0  |   1  |   1  |   0  |   1  |
-------------------------------------------------------------

-------------------------------------------------------------
 Data Structure   |  [0] |  [1] |  [2] |  [3] |  [4] |  [5] |
-------------------------------------------------------------
       DIST[]     |   0  |   5  |   1  |   2  |   3  |   4  |
-------------------------------------------------------------
        MST[]     |  -1  |   2  |   0  |   5  |   1  |   2  |
-------------------------------------------------------------
       VISITED[]  |   1  |   1  |   1  |   1  |   0  |   1  |
-------------------------------------------------------------

-------- ------
Edge     Weight
-------- ------
[2]->[1]     5
[0]->[2]     1
[5]->[3]     2
[1]->[4]     3
[2]->[5]     4
--------------
```