Code :

```
#include<iostream>
using namespace std;

#define ROW 10
#define COL 10
#define infi 9999
class MSTree {
   int graph[ROW][COL], nodes;
public:
   void createGraph();
   void primsAlgo();
};

void MSTree::createGraph() {
   int i, j;
   cout << "Enter Total Offices: ";
   cin >> nodes;
   cout << "\nEnter Adjacency Matrix: \n";
   for (i = 0; i < nodes; i++) {
      for (j = i; j < nodes; j++) {
         cout << "Enter distance between " << i << " and " << j << endl;
         cin >> graph[i][j];
         graph[j][i] = graph[i][j];
      }
   }

   for (i = 0; i < nodes; i++) {
      for (j = 0; j < nodes; j++) {
         if (graph[i][j] == 0)
            graph[i][j] = infi;    //fill infinity where path is not present
      }
   }
}
void MSTree::primsAlgo() {
   int selected[ROW], i, j, ne=0;
```

```
    int zero = 0, one = 1, min = 0, x, y;
    int cost = 0;
    for (i = 0; i < nodes; i++)
        selected[i] = zero;

    selected[0] = one;        //starting vertex is always node-0

    while (ne < nodes - 1) {
        min = infi;
        for (i = 0; i < nodes; i++) {
            if (selected[i] == one) {
                for (j = 0; j < nodes; j++) {
                    if (selected[j] == zero) {
                        if (min > graph[i][j]) {
                            min = graph[i][j];
                            x = i;
                            y = j;
                        }
                    }
                }
            }
        }
        selected[y] = one;
        cout << "\n" << x << " --> " << y;
        cost += graph[x][y];
        ne++;
    }
    cout << "\nTotal cost is: " << cost << endl;
}
int main() {
    MSTree MST;
    cout << "\nPrims Algorithm to connect several offices\n";
    MST.createGraph();
    MST.primsAlgo();
}
```

Output :

```
Prims Algorithm to connect several offices
Enter Total Offices: 9
Enter Adjacency Matrix:
Enter distance between 0 and 0
0
Enter distance between 0 and 1
4
Enter distance between 0 and 2
0
Enter distance between 0 and 3
0
Enter distance between 0 and 4
0
Enter distance between 0 and 5
0
Enter distance between 0 and 6
0
Enter distance between 0 and 7
8
Enter distance between 0 and 8
0
Enter distance between 1 and 1
0
Enter distance between 1 and 2
8
Enter distance between 1 and 3
0
Enter distance between 1 and 4
0
Enter distance between 1 and 5
0
Enter distance between 1 and 6
0
Enter distance between 1 and 7
```

```
11
Enter distance between 1 and 8
0
Enter distance between 2 and 2
0
Enter distance between 2 and 3
7
Enter distance between 2 and 4
0
Enter distance between 2 and 5
4
Enter distance between 2 and 6
0
Enter distance between 2 and 7
0
Enter distance between 2 and 8
2
Enter distance between 3 and 3
0
Enter distance between 3 and 4
9
Enter distance between 3 and 5
14
Enter distance between 3 and 6
0
Enter distance between 3 and 7
0
Enter distance between 3 and 8
0
Enter distance between 4 and 4
0
Enter distance between 4 and 5
10
Enter distance between 4 and 6
0
Enter distance between 4 and 7
0
```

```
Enter distance between 4 and 8
0
Enter distance between 5 and 5
0
Enter distance between 5 and 6
2
Enter distance between 5 and 7
0
Enter distance between 5 and 8
0
Enter distance between 6 and 6
0
Enter distance between 6 and 7
1
Enter distance between 6 and 8
6
Enter distance between 7 and 7
0
Enter distance between 7 and 8
7
Enter distance between 8 and 8
0
0 --> 1
0 --> 7
7 --> 6
6 --> 5
5 --> 2
2 --> 8
2 --> 3
3 --> 4
Total cost is: 37
```