S.No: 18	Exp. Name: ***Prim's Algorithm***	Date:2025-10-20

Aim:

Given an **adjacency matrix** of a weighted and undirected graph **G**. Find the cost of the minimum spanning tree that covers all the nodes using Prim's Algorithm. Take the elements of the adjacency matrix as a string of integers and spaces.

Constraints:

- 0 < no_of_vertices <= 100

Note:

- The elements of the matrix will always be **N * N**
- Always start from the first node to find the minimum cost
- Take the nodes names as 0,1,2,3,....,N always

Sample Test Case-1:
Input:
4
0 1 2 4 1 0 0 3 2 0 0 5 4 3 5 0

Output:
6

Sample Test Case-2:
Input:
5
0 3 0 7 8 3 0 1 4 0 0 1 0 2 0 7 4 2 0 3 8 0 0 3 0

Output:
9

Source Code:

CTC36469.c

```
#include <stdio.h>
#include <limits.h>
int main(){
   int n;
   scanf("%d", &n);
   int graph[100][100];
   int i, j, k = 0;
   for(i = 0; i<n; i++){
      for(j = 0; j<n; j++){
         scanf("%d", &graph[i][j]);
      }
   }
   int selected[100] = {0};
   selected[0] = 1;

   int edges = 0;
   int total_cost = 0;
```

```
    while(edges < n-1){
        int min = INT_MAX;
        int x=0, y=0;
        for(i = 0; i<n; i++){
            if(selected[i]){
                for(j = 0; j<n; j++){
                    if(!selected[j]&&graph[i][j]&&graph[i][j]<min){
                        min = graph[i][j];
                        x = i;
                        y = j;

                    }
                }
            }
        }
        selected[y] = 1;
        total_cost += graph[x][y];
        edges++;
    }
    printf("%d\n", total_cost);

    return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
4
0 1 2 4 1 0 0 3 2 0 0 5 4 3 5 0
6

Test Case - 2
User Output
5
0 3 0 7 8 3 0 1 4 0 0 1 0 2 0 7 4 2 0 3 8 0 0 3 0
9