```c
#include<stdio.h>
#include <stdlib.h>

#define MAX 100
#define MAXCOST 999

int graph[MAX][MAX];

int Prim(int graph[][MAX], int n)
{
    /* lowcost[i]记录以i为终点的边的最小权值，当lowcost[i]=0时表示终点i加入生成树 */
    int lowcost[MAX];

    /* mst[i]记录对应lowcost[i]的起点，当mst[i]=0时表示起点i加入生成树 */
    int mst[MAX];

    int i, j, min, minid, sum = 0;

    /* 默认选择1号节点加入生成树，从2号节点开始初始化 */
    for (i = 2; i <= n; i++)
    {
        /* 最短距离初始化为其他节点到1号节点的距离 */
        lowcost[i] = graph[1][i];

        /* 标记所有节点的起点皆为默认的1号节点 */
        mst[i] = 1;
    }

    /* 标记1号节点加入生成树 */
    mst[1] = 0;

    /* n个节点至少需要n-1条边构成最小生成树 */
    for (i = 2; i <= n; i++)
    {
        min = MAXCOST;
        minid = 0;

        /* 找满足条件的最小权值边的节点minid */
        for (j = 2; j <= n; j++)
        {
            /* 边权值较小且不在生成树中 */
            if (lowcost[j] < min && lowcost[j] != 0)
            {
                min = lowcost[j];
                minid = j;
            }
        }
        /* 输出生成树边的信息:起点，终点，权值 */
        printf("(%c --> %c):%d\n", mst[minid] + 'A' - 1, minid + 'A' - 1, min);

        /* 累加权值 */
        sum += min;

        /* 标记节点minid加入生成树 */
        lowcost[minid] = 0;

        /* 更新当前节点minid到其他节点的权值 */
        for (j = 2; j <= n; j++)
        {
```

```
            /* 发现更小的权值 */
            if (graph[minid][j] < lowcost[j])
            {
                /* 更新权值信息 */
                lowcost[j] = graph[minid][j];

                /* 更新最小权值边的起点 */
                mst[j] = minid;
            }
        }
    }
    /* 返回最小权值和 */
    return sum;
}

int main()
{
    int i,j,k,m,n;
    int cost;
    char chx,chy;
    printf("请输入顶点数和边的数目：\n");
    /* 读取节点和边的数目 */
    scanf("%d%d", &m, &n);
    getchar();

    /* 初始化图，所有节点间距离为无穷大 */
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < m; j++)
        {
            graph[i][j] = MAXCOST;
        }
    }

    printf("请输入边的信息：\n");
    printf("例如'A B 7'\n");
    /* 读取边信息 */
    for (k = 0; k < n; k++)
    {
        printf("第%d条边:",k+1);
        scanf("%c %c %d", &chx, &chy, &cost);
        getchar();
        i = chx - 'A' + 1;
        j = chy - 'A' + 1;
        graph[i][j] = cost;
        graph[j][i] = cost;
    }
    printf("得到的邻接矩阵如下:\n");
    for(int a=0;a<m;a++)
    {
        for(int b=0;b<m;b++)
        {
            printf("%d\t",graph[a][b]);
        }
        printf("\n\n");
    }
    printf("已得到最小生成树：\n");
    /* 求解最小生成树 */
    cost = Prim(graph, m);
    /* 输出最小权值和 */
    printf("最小权值和:%d\n", cost);
```

```
    //system("pause");
    return 0;
}
```