main.c

Run

```c
#192210211
#include <stdio.h>
#include <limits.h>
#define vertices 5
int minimum_key(int k[], int mst[])
{
    int minimum  = INT_MAX, min,i;
    for (i = 0; i < vertices; i++)
        if (mst[i] == 0 && k[i] < minimum )
            minimum = k[i], min = i;
    return min;
}
void prim(int g[vertices][vertices])
{
    int parent[vertices];
    int k[vertices];
    int mst[vertices];
    int i, count,edge,v;
    for (i = 0; i < vertices; i++)
    {
        k[i] = INT_MAX;
        mst[i] = 0;
    }
    k[0] = 0;
    parent[0] = -1;
    root of MST*/
    for (count = 0; count < vertices-1; count++)
    {
        edge = minimum_key(k, mst);
        mst[edge] = 1;
```

Output

Clear

```
/tmp/dKzoRx1qjh.o
Edge      Weight
 3 <-> 1    4
 0 <-> 2    3
 2 <-> 3    2
 3 <-> 4    1
```

main.c

Run

```
    k[0] = 0;
    parent[0] = -1;
    root of MST*/
    for (count = 0; count < vertices-1; count++)
    {
        edge = minimum_key(k, mst);
        mst[edge] = 1;
        for (v = 0; v < vertices; v++)
        {
            if (g[edge][v] && mst[v] == 0 && g[edge][v] <  k[v])
            {
                parent[v]  = edge, k[v] = g[edge][v];
            }
        }
     }
     printf("\n Edge \t  Weight\n");
     for (i = 1; i < vertices; i++)
     printf(" %d <-> %d    %d \n", parent[i], i, g[i][parent[i]]);
}
int main()
{
    int g[vertices][vertices] = {{0, 0, 3, 0, 0},
                                 {0, 0, 10, 4, 0},
                                 {3, 10, 0, 2, 6},
                                 {0, 4, 2, 0, 1},
                                 {0, 0, 6, 1, 0},
                                 };
    prim(g);
    return 0;
}
```

Output

Clear

```
/tmp/dKzoRx1qjh.o
Edge      Weight
 3 <-> 1    4
 0 <-> 2    3
 2 <-> 3    2
 3 <-> 4    1
```