Q-Write a program to implement Prims Algorithm using Disjoint Sets.

```
#include <bits/stdc++.h>
using namespace std;
const int MAXN = 10;
int parent[MAXN], rank_arr[MAXN];
int find(int x) {
if (parent[x] == x) return x;
return parent[x] = find(parent[x]);
}
void union_sets(int x, int y) {
    x = find(x);
    y = find(y);
if (x == y) return;
if (rank_arr[x] < rank_arr[y]) swap(x, y);
parent[y] = x;
if (rank_arr[x] == rank_arr[y]) rank_arr[x]++;
}
int main() {
int n, m;
cout<<"Enter no. of vertices ";
cin>>n;
cout<<"Enter no. of edges ";
cin>>m;
for (int i = 1; i <= n; i++) {
parent[i] = i;
```

```
        rank_arr[i] = 1;
    }
int u, v, w;
for (int i = 1; i <= m; i++) {
cin>> u >> v >> w;
        union_sets(u, v);
    }
int mst_weight = 0;
for (int i = 1; i <= n; i++) {
if (parent[i] == i) {
            mst_weight += w;
cout<< "Edge: " << i << " - " << parent[i] << " Weight: " << w << endl;
        }
    }
cout<< "Total MST Weight: " << mst_weight << endl;
return 0;
}
```

Q-Introduction to Linux – Write a program to print Hello World.

```
#include <bits/stdc++.h>
using namespace std;
const int MAXN = 10;
int parent[MAXN], rank_arr[MAXN];
```

```
int find(int x) {
if (parent[x] == x) return x;
return parent[x] = find(parent[x]);
}
void union_sets(int x, int y) {
    x = find(x);
    y = find(y);
if (x == y) return;
if (rank_arr[x] < rank_arr[y]) swap(x, y);
parent[y] = x;
if (rank_arr[x] == rank_arr[y]) rank_arr[x]++;
}
int main() {
int n, m;
cout<<"Enter no. of vertices ";
cin>>n;
cout<<"Enter no. of edges ";
cin>>m;
for (int i = 1; i <= n; i++) {
parent[i] = i;
        rank_arr[i] = 1;
    }
int u, v, w;
for (int i = 1; i <= m; i++) {
cin>> u >> v >> w;
```

```
        union_sets(u, v);
    }
int mst_weight = 0;
for (int i = 1; i <= n; i++) {
if (parent[i] == i) {
            mst_weight += w;
cout<< "Edge: " << i << " - " << parent[i] << " Weight: " << w << endl;
        }
    }
cout<< "Total MST Weight: " << mst_weight << endl;
return 0;
}
```