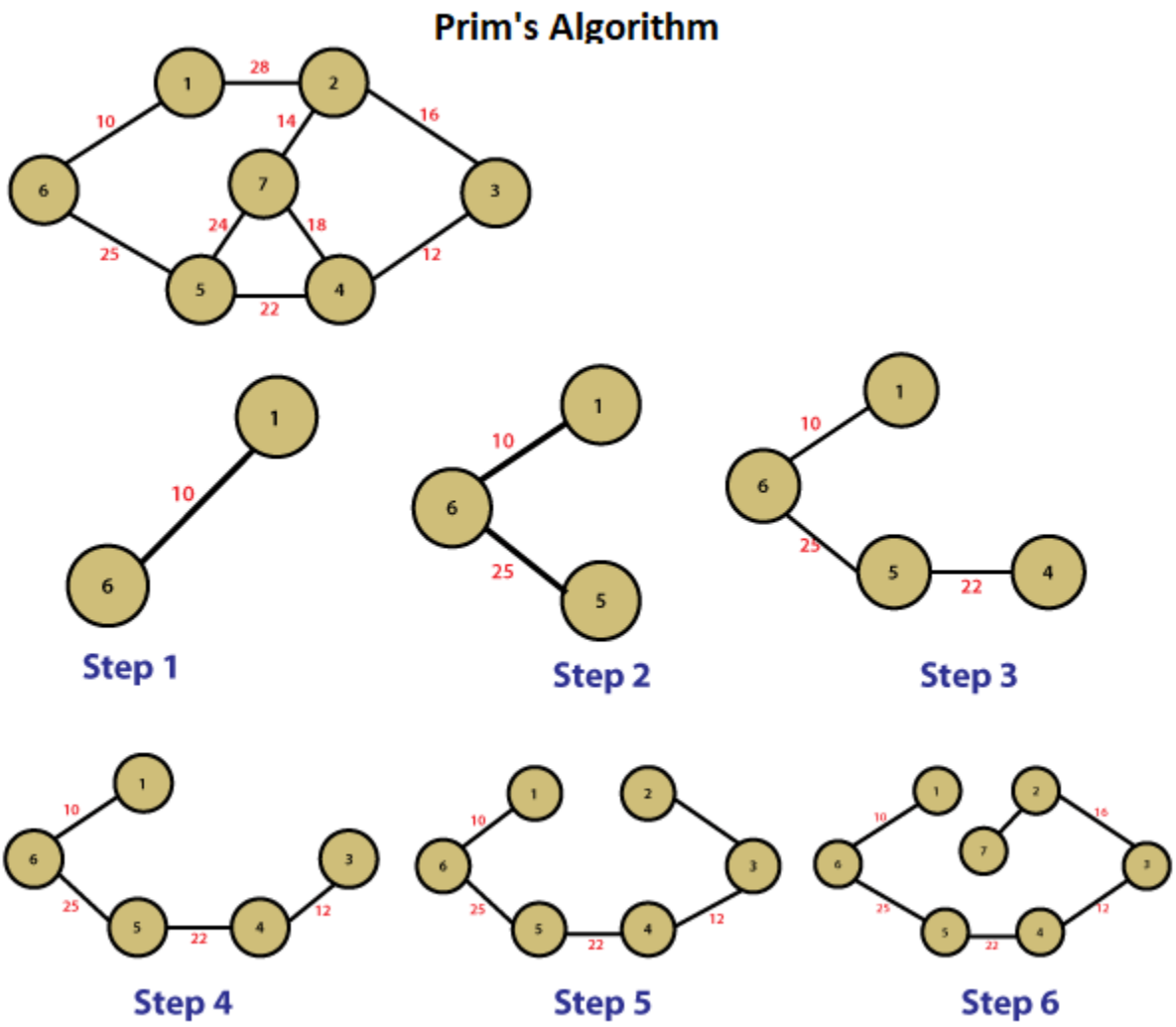


```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>
```

```
int weight[20][20],visited[20],d[20],p[20],v, e;
void creategraph()
	{
		int i,j,a,b,w;
		printf("Enter the number of vertices");
		scanf("%d",&v);
		printf("Enter the number of edges");
		scanf("%d",&e);
		for ( i=1;i<=v;i++)
			for( j=1;j<=v;j++)
				weight[i][j]=0;

		for (i=1;i<=v;i++)
			{
			p[i]=visited[i]=0;
			d[i]=32767;
			}
		for ( i=1;i<=e;i++)
		{
		printf ("\nEnter edge a,b and weight w :");
		scanf("%d %d %d" ,&a,&b,&w);
		weight[a][b]=weight[b][a]=w;
		}

	}
```

```
void algo ()
        {
         creategraph();
          int current,total,mincost,i;
          current=1;d[current]=0;
           total=1;
           visited[current]=1;
           while(total!=v)
                    {
                      for (i=1;i<=v;i++)
                       {
                      if(weight[current][i]!=0)
                       if(visited[i]==0)
                       if(d[i]>weight[current][i])
                             {
                           d[i]=weight[current][i];
                            p[i]=current;
                           }
                      }
        mincost=32767;
          for (i=1;i<=v;i++)
            {
               if(visited[i]==0)
               if(d[i]<mincost)
                    {
```

```
                mincost=d[i];
                current=i;
                  }
            }
        visited[current]=1;
        total++;
    }
        mincost=0;
        for(i=1;i<=v;i++)
        mincost=mincost+d[i];
        printf ("\n Minimum cost=%d", mincost);
        printf("\n Minimum Spanning tree is");
        for(i=1;i<=v;i++)
        printf("\n vertex %d is connected to %d",i,p[i]);
}

void main()
{
      algo();
}
```

Enter number of vertices :5

Enter number of Edges :7

Enter edge a,b and weight w :1

2

2

Enter edge a,b and weight w :1

5

8

Enter edge a,b and weight w :2

3

6

Enter edge a,b and weight w :2

4

12

Enter edge a,b and weight w :2

5

7

Enter edge a,b and weight w :4

5

4

Enter edge a,b and weight w :3

```
4

10


 Minimum cost=19

 Minimum Spanning tree is

 vertex1is connected to0

 vertex2is connected to1

 vertex3is connected to2

 vertex4is connected to5

 vertex5is connected to2

*/



int i,j,k,a,b,u,v,n,ne=1;

int min,mincost=0,cost[9][9],parent[9];

int find(int);

int uni(int,int);

void main()

{

        //clrscr();

        printf("\n\tImplementation of Kruskal's algorithm\n");

        printf("\nEnter the no. of vertices:");

        scanf("%d",&n);

        printf("\nEnter the cost adjacency matrix:\n");

        for(i=1;i<=n;i++)
```

```
{
        for(j=1;j<=n;j++)
        {
                scanf("%d",&cost[i][j]);
                if(cost[i][j]==0)
                        cost[i][j]=999;
        }
}
printf("The edges of Minimum Cost Spanning Tree are\n");
while(ne < n)
{
        for(i=1,min=999;i<=n;i++)
        {
                for(j=1;j <= n;j++)
                {
                        if(cost[i][j] < min)
                        {
                                min=cost[i][j];
                                a=u=i;
                                b=v=j;
                        }
                }
        }
        u=find(u);
        v=find(v);
```

```
                if(uni(u,v))
                {
                        printf("%d edge (%d,%d) =%d\n",ne++,a,b,min);
                        mincost +=min;
                }
                cost[a][b]=cost[b][a]=999;
        }
        printf("\n\tMinimum cost = %d\n",mincost);
        getch();
}
int find(int i)
{
        while(parent[i])
        i=parent[i];
        return i;
}
int uni(int i,int j)
{
        if(i!=j)
        {
                parent[j]=i;
                return 1;
        }
        return 0;
}
```

0	4	1	3
4	0	0	2
1	0	0	3
3	2	3	0