```
1115
         //Prims
1116
         #include<stdio.h>
         #include<stdlib.h>
1117
         int a,b,u,t,m,n,i,j;
1118
1119
         int v[10], c=1, min=0, cost[10][10];
1120
         int main() {
1121
             printf("Enter the number of vertices :- ");
1122
             scanf ("%d", &n);
1123
             printf("\nEnter the adjacency matrix:\n");
             for (i=1;i<=n;i++) {</pre>
1124
                  for (j=1; j<=n; j++) {
1125
1126
                      scanf("%d", &cost[i][j]);
1127
                      if(cost[i][j]==0)
1128
                           cost[i][j]=999;
1129
1130
             v[1]=1;
1131
1132
             while (c<n) {
                  for (i=1, m=999; i<=n; i++) {
1133
1134
                      for (j=1;j<=n;j++) {</pre>
1135
                           if(cost[i][j]<m){</pre>
1136
                               if(v[i]!=0){
1137
                                   m=cost[i][j];
1138
                                   a=u=i;
1139
                                   b=t=j;
1140
```

```
b=t=j;
1139
1140
1141
1142
1143
                 if(v[u]==0 || v[t]==0) {
1144
                     printf("\nEdge %d:(%d %d) cost:%d",c++,a,b,m);
1145
1146
                     min+=m;
1147
                     v[b]=1;
1148
1149
                 cost[a][b]=cost[b][a]=999;
1150
             printf("\nMinimun cost=%d\n",min);
1151
1152
             return 0;
1153
1154
1155
```

```
Enter the number of vertices :- 3
Enter the adjacency matrix:
0 2 5
200
5 0 0
Edge 1:(1 2) cost:2
Edge 2:(1 3) cost:5
Minimun cost=7
Process returned 0 (0x0) execution time : 21.159 \text{ s}
Press any key to continue.
```

```
1158
          /Kruskal
1159
         #include<stdio.h>
1160
         #include<stdlib.h>
1161
         int i, j, k, a, b, u, v, n, ne=1;
1162
         int min, mincost=0, cost[10][10], parent[10];
1163
         int main()
1164
             printf("Enter the no. of vertices :- ");
1165
1166
             scanf ("%d", &n);
1167
             printf("\nEnter the adjacency matrix :\n");
             for (i=1; i<=n; i++) {
1168
1169
                  for (j=1; j<=n; j++) {
1170
                      scanf("%d", &cost[i][j]);
1171
                      if(cost[i][j]==0)
                           cost[i][j]=999;
1172
1173
1174
1175
             while (ne < n) {
1176
                  for (i=1, min=999; i<=n; i++) {
1177
                      for(j=1; j <= n; j++) {
                           if(cost[i][j] < min) {
1178
1179
                               min=cost[i][j];
1180
                               a=u=i;
1181
                               b=v=j;
1182
1183
1184
1185
                  while (parent[u])
1186
                      u=parent[u];
```

```
while (parent[u])
1185
1186
                     u=parent[u];
                 while (parent[v])
1187
1188
                     v=parent[v];
1189
                 if(u!=v){
                     printf("Edge %d:%d,%d) Cost:%d\n",ne++,a,b,min);
1190
1191
                     mincost+=min;
1192
                     parent[v]=u;
1193
1194
                 cost[a][b]=cost[b][a]=999;
1195
             printf("\nMinimum cost = %d\n", mincost);
1196
1197
             return 0;
1198
1199
1200
1201
1202
1203
```

```
Enter the adjacency matrix :
031600
3 0 5 0 3 0
150564
6 0 5 0 0 2
036006
004260
Edge 1:(1,3) Cost:1
Edge 2:(4,6) Cost:2
Edge 3:(1,2) Cost:3
Edge 4:(2,5) Cost:3
Edge 5:(3,6) Cost:4
Minimum cost = 13
Process returned 0 (0x0) execution time : 48.654 s
Press any key to continue.
```

Enter the no. of vertices :- 6