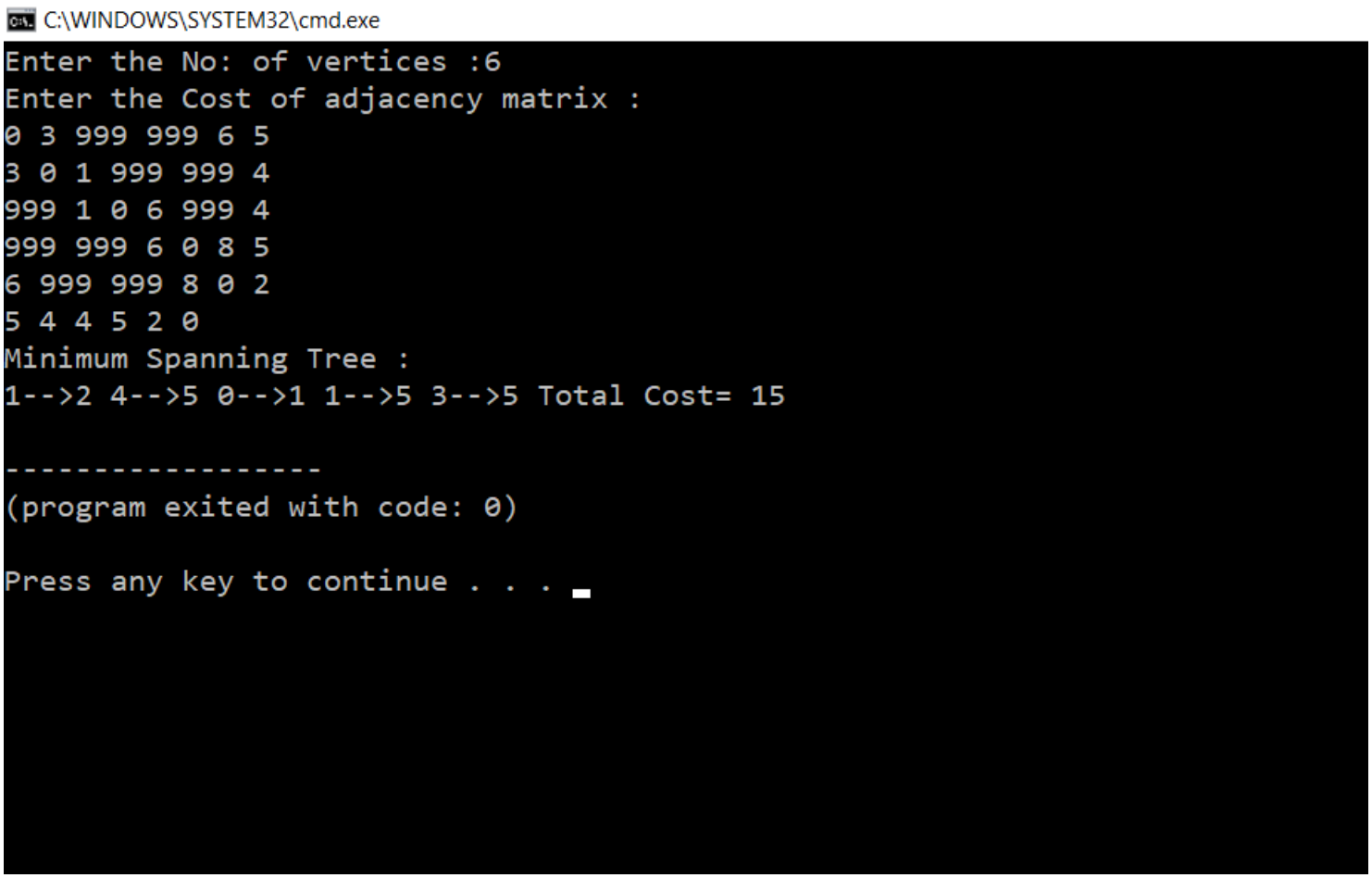
```c
#include<stdio.h>
#include<process.h>


void kruskals();
int cost[10][10],n,min,i,j,sum;
int count,k,u,v,parent[10];
int t[10][10];
void union_ij(int,int);
int find(int);
int  main()
{
  printf("Enter the No: of vertices :");
  scanf("%d",&n);
  printf("Enter the Cost of adjacency matrix :\n");
  for(i=0;i<n;i++)
 {
  for(j=0;j<n;j++)
  {
   scanf("%d",&cost[i][j]);
  }
 }
  kruskals();
  return 0;
}

void kruskals()
{
  count=0;
  k=0;
  sum=0;
  for(i=0;i<n;i++)
  parent[i]=i;
  while(count!=n-1)
 {
  min=999;
  for(i=0;i<n;i++)
 {
```

```
 for(i=0;i<n;i++)
 parent[i]=i;
 while(count!=n-1)
{
 min=999;
 for(i=0;i<n;i++)
{
 for(j=0;j<n;j++)
 {
  if(cost[i][j]<min && cost[i][j]!=0)
  {
    min=cost[i][j];
    u=i;
    v=j;
  }
 }
}
 i=find(u);
 j=find(v);
 if(i!=j)
{
 t[k][0]=u;
 t[k][1]=v;
 k++;
 count++;
 sum=sum+cost[u][v];
 union_ij(i,j);
}
 cost[u][v]=cost[v][u]=999;
}
 printf("Minimum Spanning Tree :\n");
 for(i=0;i<n-1;i++)
{
 printf("%d-->%d ",t[i][0],t[i][1]);
}
 printf("Total Cost= %d", sum);
}
```

```
 }
  i=find(u);
  j=find(v);
  if(i!=j)
 {
  t[k][0]=u;
  t[k][1]=v;
  k++;
  count++;
  sum=sum+cost[u][v];
  union_ij(i,j);
 }
  cost[u][v]=cost[v][u]=999;
 }
  printf("Minimum Spanning Tree :\n");
  for(i=0;i<n-1;i++)
 {
  printf("%d-->%d ",t[i][0],t[i][1]);
 }
  printf("Total Cost= %d", sum);
}

void union_ij(int i, int j)
{
  if(i<j)
  parent[j]=i;
  else
  parent[i]=j;
}

int find(int v)
{
  while(parent[v]!=v)
  v=parent[v];
  return v;
}
```


C:\WINDOWS\SYSTEM32\cmd.exe
Enter the No: of vertices :6
Enter the Cost of adjacency matrix :
0 3 999 999 6 5
3 0 1 999 999 4
999 1 0 6 999 4
999 999 6 0 8 5
6 999 999 8 0 2
5 4 4 5 2 0
Minimum Spanning Tree :
1-->2 4-->5 0-->1 1-->5 3-->5 Total Cost= 15

(program exited with code: 0)
Press any key to continue . . .