

```

1  #include<iostream>
2  using namespace std;
3  const int n=8; //顶点数
4  const int e=10; //边数
5  bool visited[n+1];
6  typedef int elemtype;
7
8  class graph{
9  public:
10 elemtype v[n+1];
11 int arcs[n+1][n+1];
12
13 void creatadj(graph g);
14 void dfs(graph g,int i);
15 };
16
17 void graph::creatadj(graph g){
18 int i,j,k;
19 for(k=1;k<=n;k++)
20 {
21     cin>>g.v[k];
22 }
23
24 for(i=1;i<=n;i++)
25     for(j=1;j<=n)
26 {
27     g.arcs[i][j]=0;
28     g.arcs[j][i]=0;
29 }
30 for(k=1;k<=e;k++)
31 {
32     cin>>i>>j;
33     g.arcs[i][j]=1;
34     g.arcs[j][i]=1;
35 }
36 }
37
38 void graph::dfs(graph g,int i){
39 int j;
40 cout<<g.v[i];
41 visited[i]=true;
42 for(j=1;j<=n;j++)
43 {
44     if((g.arcs[i][j]==1)&&(!visited[j]))
45         dfs(g,j);
46 }
47 }
48
49 int main()
50 {
51     graph g;
52     g.creatadj(g);
53     for(int i=1;i<=n;i++)
54         visited[i]=false;
55     g.dfs(g,1);
56     return 0;
57 }

```