```cpp
//日期: 2018/ 时间:
#include <stdio.h>
#include <stdlib.h>
#include <queue>
using namespace std;

const int maxv = 1000;  //最大顶点数
const int INF = 1000000000; //INF为很大的数

//#define p1
#define p2

//**************************邻接矩阵**************************************
#ifdef p1
int n,G[maxv][maxv];        //n为顶点数
bool inq[maxv] = {false};   //若顶点i曾入过队，则inq[i]==true.初值为false

void BFS(int u){            //遍历u所在的连通块
    queue<int> q;           //定义队列q
    q.push(u);              //将初试节点u入队
    inq[u] = true;          //设置u已被加入过队列
    while(!q.empty()){      //只要队列非空
        int u = q.front();
        q.pop();
        for(int v=0;v<n;v++){
            //入过u的邻接点v未曾加入过队列
            if(inq[v] == false && G[u][v]!=INF){
                q.push(v);
                inq[v] = true;
            }
        }
    }

}
void BFSTrave(){            //遍历图G
    for(int u=0;u<n;n++){   //枚举所有顶点
        if(inq[u] == false){//如果u未曾加入过队列
            BFS(u);
        }
    }
}
#endif

//**************************邻接表**************************************
#ifdef p2
vector<int> adj[maxv];      //图G的邻接表
int n;                      //n为顶点数
bool inq[maxv] = {false};

void BFS(int u){            //遍历单个连通块
    queue<int> q;
    q.push(u);
    inq[u] = true;
    while(!q.empty()){
        int u = q.front();
        q.pop();
```

```
57          for(int i=0;i<adj[u].size();i++){
58              int v = adj[u][i];
59              if(inq[v] == false){
60                  q.push(v);
61                  inq[v] = true;
62              }
63          }
64      }
65  }
66
67  void BFSTrave(){
68      for(int u=0;u<n;u++){
69          if(inq[u] == false){
70              BFS(u);
71          }
72      }
73  }
74
75  #endif
76
77  int main(){
78
79
80      return 0;
81  }
82
83
```