

## 青少年软件编程（C 语言）等级考试试卷（七级）

分数：100 题数：4

### 一、编程题(共 4 题，共 100 分)

#### 1. 玩转二叉树

战争中保持各个城市间的连通性非常重要。本题要求你编写一个报警程序，当失去一个城市导致国家被分裂为多个无法连通的区域时，就发出红色警报。注意：若该国本来就不完全连通，是分裂的  $k$  个区域，而失去一个城市并不改变其他城市之间的连通性，则不要发出警报。

时间限制：5000

内存限制：65536

#### 输入

输入在第一行给出两个整数  $N$  ( $0 < N \leq 500$ ) 和  $M$  ( $\leq 5000$ )，分别为城市个数（于是默认城市从 0 到  $N-1$  编号）和连接两城市的通路条数。随后  $M$  行，每行给出一条通路所连接的两个城市的编号，其间以 1 个空格分隔。在城市信息之后给出被攻占的信息，即一个正整数  $K$  和随后的  $K$  个被攻占的城市的编号。注意：输入保证给出的被攻占的城市编号都是合法的且无重复，但并不保证给出的通路没有重复。

#### 输出

对每个被攻占的城市，如果它会改变整个国家的连通性，则输出“Red Alert: City  $k$  is lost!”，其中  $k$  是该城市的编号；否则只输出“City  $k$  is lost.”即可。如果该国失去了最后一个城市，则增加一行输出“Game Over.”。

#### 样例输入

```
5 4
0 1
1 3
3 0
0 4
5
1 2 0 4 3
```

#### 样例输出

```
City 1 is lost.
City 2 is lost.
Red Alert: City 0 is lost!
City 4 is lost.
```

City 3 is lost.

试题编号：20240622-7-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
const int maxn = 505;
```

```
bool lost[maxn], vst[maxn];
```

```
vector<int> G[maxn];
```

```
int n, m, ans;
```

```
void dfs(int u)
```

```
{
```

```
    vst[u] = true;
```

```
    for(int & x : G[u]) if(!lost[x] && !vst[x]) dfs(x);
```

```
}
```

```
int test()
```

```
{
```

```
    int res = 0;
```

```
    for(int i = 0; i < n; i++) vst[i] = false;
```

```
    for(int i = 0; i < n; i++)
```

```
    {
```

```
        if(lost[i]) continue;
```

```
        if(!vst[i]) ++res, dfs(i);
```

```
    }
```

```
    return res;
```

```
}
```

```
int main()
```

```
{
```

```
    #ifdef MyTest
```

```
        freopen("Sakura.txt", "r", stdin);
```

```
#endif
```

```
    cin >> n >> m;
```

```
    for(int i = 0; i < m; i++)
```

```
    {
```

```
        int x, y;
```

```
        cin >> x >> y;
```

```

        G[x].push_back(y);
        G[y].push_back(x);
    }
    int k;
    cin >> k;
    ans = test();
    for(int i = 0; i < k; i++)
    {
        int x;
        cin >> x;
        lost[x] = true;
        int now = test();
        if(now > ans) printf("Red Alert: City %d is lost!\n", x);
        else printf("City %d is lost.\n", x);
        ans = now;
    }
    if(k == n) puts("Game Over.");
    return 0;
}

```

考生得分：25

是否评分：已评分

评价描述：

## 2. 红色警报

战争中保持各个城市间的连通性非常重要。本题要求你编写一个报警程序，当失去一个城市导致国家被分裂为多个无法连通的区域时，就发出红色警报。注意：若该国本来就不完全连通，是分裂的  $k$  个区域，而失去一个城市并不改变其他城市之间的连通性，则不要发出警报。

时间限制：5000

内存限制：65536

### 输入

输入在第一行给出两个整数  $N$  ( $0 < N \leq 500$ ) 和  $M$  ( $\leq 5000$ )，分别为城市个数（于是默认城市从 0 到  $N-1$  编号）和连接两城市的通路条数。随后  $M$  行，每行给出一条通路所连接的两个城市的编号，其间以 1 个空格分隔。在城市信息之后给出被攻占的信息，即一个正整数  $K$  和随后的  $K$  个被攻占的城市的编号。注意：输入保证给出的被攻占的城市编号都是合法的且无重复，但并不保证给出的通路没有重复。

### 输出

对每个被攻占的城市，如果它会改变整个国家的连通性，则输出“Red Alert: City  $k$  is lost!”，其中  $k$  是该城市的编号；否则只输出“City  $k$  is lost.”即可。如果该国失去了最后一个城市，则增加一行输出“Game Over.”。

### 样例输入

```
5 4
0 1
1 3
3 0
0 4
5
1 2 0 4 3
```

### 样例输出

```
City 1 is lost.
City 2 is lost.
Red Alert: City 0 is lost!
City 4 is lost.
City 3 is lost.
```

试题编号 : 20240622-7-02

试题类型 : 编程题

标准答案 :

试题难度 : 一般

试题解析 :

展示地址 : [点击浏览](#)

考生答案 :

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
const int maxn = 505;
```

```
bool lost[maxn], vst[maxn];
```

```
vector<int> G[maxn];
```

```
int n, m, ans;
```

```
void dfs(int u)
```

```
{
```

```
    vst[u] = true;
```

```
    for(int & x : G[u]) if(!lost[x] && !vst[x]) dfs(x);
```

```
}
```

```
int test()
```

```
{
```

```

    int res = 0;
    for(int i = 0; i < n; i++) vst[i] = false;
    for(int i = 0; i < n; i++)
    {
        if(lost[i]) continue;
        if(!vst[i]) ++res, dfs(i);
    }
    return res;
}
int main()
{
    #ifdef MyTest
        freopen("Sakura.txt", "r", stdin);
    #endif
    cin >> n >> m;
    for(int i = 0; i < m; i++)
    {
        int x, y;
        cin >> x >> y;
        G[x].push_back(y);
        G[y].push_back(x);
    }
    int k;
    cin >> k;
    ans = test();
    for(int i = 0; i < k; i++)
    {
        int x;
        cin >> x;
        lost[x] = true;
        int now = test();
        if(now > ans) printf("Red Alert: City %d is lost!\n", x);
        else printf("City %d is lost.\n", x);
        ans = now;
    }
    if(k == n) puts("Game Over.");
    return 0;
}

```

考生得分：25

是否评分：已评分

评价描述：

3. 到底爱不爱我

古代少女有了心上人时，会悄悄折一条树枝，揪那枝上的叶子，揪一片叶子念一句“爱我”，再揪一片念一句“不爱我”……这样揪落最后一叶子的时候，看看是停在“爱”还是“不爱”。

但聪明的慧娘一眼洞穿，只要数一下叶子有多少片，根据这个数字的奇偶性判断是以“爱”开始还是以“不爱”开始，就总是可以最后落在“爱”上。这个游戏顿时就变得无趣了——真的是文科生制造浪漫，理科生杀死浪漫。

于是有着工科生大脑的慧娘打算另外制作一个更有趣的浪漫游戏。她用不同植物的枝条做成了三种“情枝”：

- “专情枝”：是一根有两个分岔的树枝，只有当两个分岔上连接的枝条传过来的情话都是“爱”的时候，这根枝条的根部才传出“爱”；否则树枝根部传出的是“不爱”。
- “博爱枝”：也是一根有两个分岔的树枝，只有当两个分岔上连接的枝条传过来的情话都是“不爱”的时候，这根枝条的根部才传出“不爱”；否则树枝根部传出的都是“爱”。
- “情变枝”：是没有分岔的一根直枝，如果一端接到“爱”，另一端必须传出“不爱”；反之如果一端接到“不爱”，另一端则会传出“爱”。

慧娘将这些树枝摆放在院子里，布了一个“情阵”，从某一枝的根部开始，扩散开去，令它们根枝相连。然后她在末梢的枝杈旁随意写下“爱”或“不爱”。现在请你写个程序帮她算出来，在初始一枝的根部，她能得到“爱”还是“不爱”？

时间限制：10000

内存限制：65536

## 输入

输入在第一行中给出正整数  $N$  ( $\leq 30$ )，是慧娘制作的情枝数量。这里假设她将所有的情枝从 1 到  $N$  做好了编号。随后  $N$  行，第  $i$  行给出第  $i$  枝的描述，格式为 类型 左分枝连接的编号 右分枝连接的编号 其中 `类型` 为 1 代表专情、2 代表博爱、3 代表情变。当然如果是情变枝，则后面跟的是其唯一末端连接的情枝编号，并没有两个分枝的信息。如果一个分枝是末梢，并没有连接其它枝条，则对应编号为 0。接下来一行中给出正整数  $K$  ( $\leq 30$ )，是慧娘询问的次数。以下  $K$  行，每行给出一个由 0 和 1 组成的字符串，其中 0 表示“不爱”，1 表示“爱”——这是慧娘从左到右在每个枝杈末梢处写下的。

## 输出

对慧娘的每个询问，如果她在初始一枝的根部能得到“爱”，就输出 `Ai` ；否则输出 `BuAi` 。

样例输入

```
6
2 6 4
1 0 0
3 1
2 0 0
3 0
1 5 2
5
11111
00000
11100
10011
01100
```

样例输出

```
BuAi
Ai
Ai
BuAi
BuAi
```

提示

样例说明： 样例对应的情阵以及慧娘第 3 问的情势如图所示，其中完整的心对应 1，裂开的心对应 0。



试题编号：20240622-7-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：0

是否评分：已评分

评价描述：

#### 4. 兰州拉面派餐系统

兰州拉面是著名美食，其煮面很有讲究，不同种类的面需要煮不同的时长。拉面馆的煮面师傅的规则很简单，只要手头有煮面篮子是空闲的，就把下一份客单指定的面放到空闲篮子里煮；如果空闲的篮子不止一个，那么先放到其中编号最小的篮子里煮；如果没有篮子是空闲的，就等待。一个篮子里的面煮够了时间，师傅要准时捞出来放到该客单对应的碗里，招呼服务员端给客人；如果有多个篮子里的面同时到了捞取时间，师傅会同时捞起，但要本着先到先得的原则，按下单顺序招呼送餐。

在生意火爆的拉面馆里，煮面师傅需要很强的记忆力和计时能力，才能面对源源不断的客单，同时照顾一大堆煮面篮子而不出错。如果面的品种不太多，篮子也不太多，那一

位拉面师傅的大脑还是能应对的。但如果我们有上千种面、上万只煮面篮、数十万客单呢…… 这就需要请你帮忙，写个派餐系统来完成这个任务了。

时间限制：7000

内存限制：65536

### 输入

输入在第一行中给出 3 个正整数：N ( $\leq 10^3$ ) 为面的种类数；M ( $\leq 10^4$ ) 为煮面篮子的个数；L ( $\leq 10^5$ ) 为客单总数。于是我们假设面的种类从 1 到 N 编号、篮子从 1 到 M 编号、客单从 1 到 L 编号。第二行给出 N 个不超过 10 的正整数，第 i 个数字对应第 i 种面需要煮的时间长度（分钟）。第三行给出 L 个正整数，第 i 个数字对应第 i 份客单所点的面种 —— 这里我们将情况简化为一份客单只要一种面，并且只要一碗，所以只要给出面的种类即可。一行中数字间以空格分隔。因为面馆的生意太火爆，我们可以认为所有的客单都是在开门前就预约好的，并且是按预约的顺序递增编号的。此外，与煮面的时长相比，下面、捞面和送餐的时长是可以忽略不计的。

### 输出

首先按照送单顺序在一行中输出所有客单的编号和送餐时间，格式为 `编号:时间`。随后在下一行按照编号递增序输出每只篮子煮了多少碗面。一行的输出之间以 1 个空格分隔，行首尾不得有多余空格。

### 样例输入

```
3 5 10
3 5 8
2 1 3 3 1 1 2 1 3 2
```

### 样例输出

```
2:3 5:3 1:5 6:6 3:8 4:8 7:8 8:8 10:13 9:14
3 3 1 1 2
```

试题编号：20240622-7-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

考生答案：（此题已作答）

考生得分：0

是否评分：已评分

评价描述：