

# 寒假集训第一次测试题

测试时间：2018 年 1 月 26 日 9:00 ~ 12:00

题目名称	零食	迷宫	彩带染色	Degrees of Separation
目录	dessert	maze	color	friend
可执行文件名	dessert	maze	color	friend
输入文件名	dessert.in	maze.in	color.in	friend.in
输出文件名	dessert.out	maze.out	color.out	friend.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512MB	512MB	512MB	512MB
测试点数目	10	20	10	4
每个测试点分值	10	5	10	25

## 1 零食 (dessert)

### 【问题描述】

情人节就要到了，NULL 打算在情人节送给他的女朋友一个零食大礼包。NULL 了解他的女朋友喜欢什么口味的零食，同时他还想探索一些他女朋友可能喜欢的零食。

我们知道 NULL 买了一个容量为  $v$  的礼盒和许许多多零食。零食分为两类，一类是之前送给过女朋友的零食，一共有  $n$  种，每种有 1 个。同时，每种零食会占用空间为  $C_i$ ，而 NULL 的女朋友对每种零食的喜爱程度为  $W_i$ 。另一类是之前没有送给过女朋友的零食，一共有  $m$  种，每种有足量，同时，每种零食会占用空间为  $C_i$ ，由于不知道女朋友是否喜爱，因此没有喜爱程度。

NULL 打算这样配置他的零食大礼包，首先用第一类零食放入礼盒，以求获得最大的喜爱程度。然后用第二类零食填充剩下的空间，尽量填满这个礼盒。所以，他最多能获得多少喜爱值，礼盒最少能剩下多少空间呢？

### 【输入格式】

输入文件名为 `dessert.in`。

第一行包含三个整数  $n$ ， $m$  和  $v$ ，中间用一个空格隔开，分别代表第一类零食数、第二类零食数和礼盒的空间。

接下来  $n$  行，每行给出两个整数  $C$  和  $W$ ，中间用一个空格隔开，分别代表第一类零食的空间和对这种零食的喜爱程度。

接下来  $m$  行，每行给出一个整数  $C$ ，代表第二类零食的空间。

**【输出格式】**

输出文件名为 `dessert.out`。

输出一行，包含两个整数，用一个空格隔开，分别表示最多获得的喜爱值和在获得最多喜爱值的情况下礼盒最少能剩下的空间。

**【输入输出样例】**

```
dessert.in
3 2 10
4 5
5 7
6 4
2
5
dessert.out
12 1
```

**【数据说明】**

对于 20% 的数据， $1 \leq n \leq 20$ ， $1 \leq m \leq 20$ ， $1 \leq v \leq 500$ ，保证第一种零食即使全部获取也不会填满礼盒。对于 100% 的数据， $1 \leq n \leq 1000$ ， $1 \leq m \leq 1000$ ， $1 \leq v \leq 100000$ ， $0 \leq W_i$ 。

## 2 迷宫 (maze)

**【问题描述】**

NULL 同学最近迷上了一个迷宫游戏。在迷宫中他需要按照迷宫的规则，从迷宫的起点 (S 点) 走到迷宫的终点 (T 点)。但是由于地图的限制，在游戏过程中，NULL 同学可能会到达地图上的某些位置，使得 NULL 同学再也无法到达终点。所以 NULL 同学想让你写程序帮他计算一下，迷宫中究竟有多少个可以从起点出发到达，却无法到达终点的点。

游戏有以下规则：

- 1、S 为起点，T 为终点。
- 2、在 '.' 标记的点上，只能往下方移动。
- 3、在 '|' 标记的点上，只能往上下移动。
- 4、在 '-' 标记的点上，只能往左右移动。
- 5、在 '+' 标记的点上，可以向上下左右四个方向移动。
- 6、你不能通过 '#' 标记的点。

**【输入格式】**

第一行，包含两个整数，R 和 C ( $1 \leq R, C \leq 50$ )，分别表示迷宫的高和长。接下来有 R 行输入，每行 C 个字符，即为按照上述规则描述的迷宫。

**【输出格式】**

你需要输出一个数表示有多少个可以从起点出发到达，却无法到达终点的点。如果从当前的起点出发无法到达终点，输出“I'm stuck!”。

**【输入输出样例】**

```
maze.in
5 5
--+-+
..|#.
..|##
S-+-T
####.
maze.out
2
```

**【样例说明】**

将从起点出发可以到达，却无法到达终点的点用 X 标记，则可以得到。

```
--+-+
..|#X
..|##
S-+-T
####X
```

### 3 彩带染色 (color)

**【问题描述】**

NULL 同学有一条彩带，为了装饰自己的教室，他需要重新对这条彩带进行染色。NULL 同学每次可以将彩带上一个相邻的区间染成某同一种颜色，现在，NULL 同学想要请你帮忙计算一下把 NULL 同学的这条彩带染成需要的颜色最少需要进行多少次染色。

**【输入格式】**

有多组测试数据。

每组测试数据有两行，第一行为当前彩带的颜色，第二行为目标彩带的颜色。

**【输出格式】**

每组测试数据输出一个整数，为最少需要对彩带进行染色的次数。

**【输入输出样例】**

```

color.in
zzzzzfzzzzz
abcdefedcba
abababababab
cdcdcdcdcdcd
color.out
6
7

```

**【数据说明】**

对于每一组测试数据，彩带长度  $L \leq 100$ 。

## 4 Degrees of Separation

### Problem Description

The main socializing tool for students today is Facebook. There are many interesting computational questions connected to Facebook, such as the “degree of separation” between two people.

For example, in the diagram below, there are many different paths between Abby and Alberto. Some of these paths are:

- Abby  $\rightarrow$  Zoey  $\rightarrow$  Alberto
- Abby  $\rightarrow$  Natalie  $\rightarrow$  Zoey  $\rightarrow$  Alberto
- Abby  $\rightarrow$  George  $\rightarrow$  Ali  $\rightarrow$  Kara  $\rightarrow$  Richardo  $\rightarrow$  Jeff  $\rightarrow$  Alberto

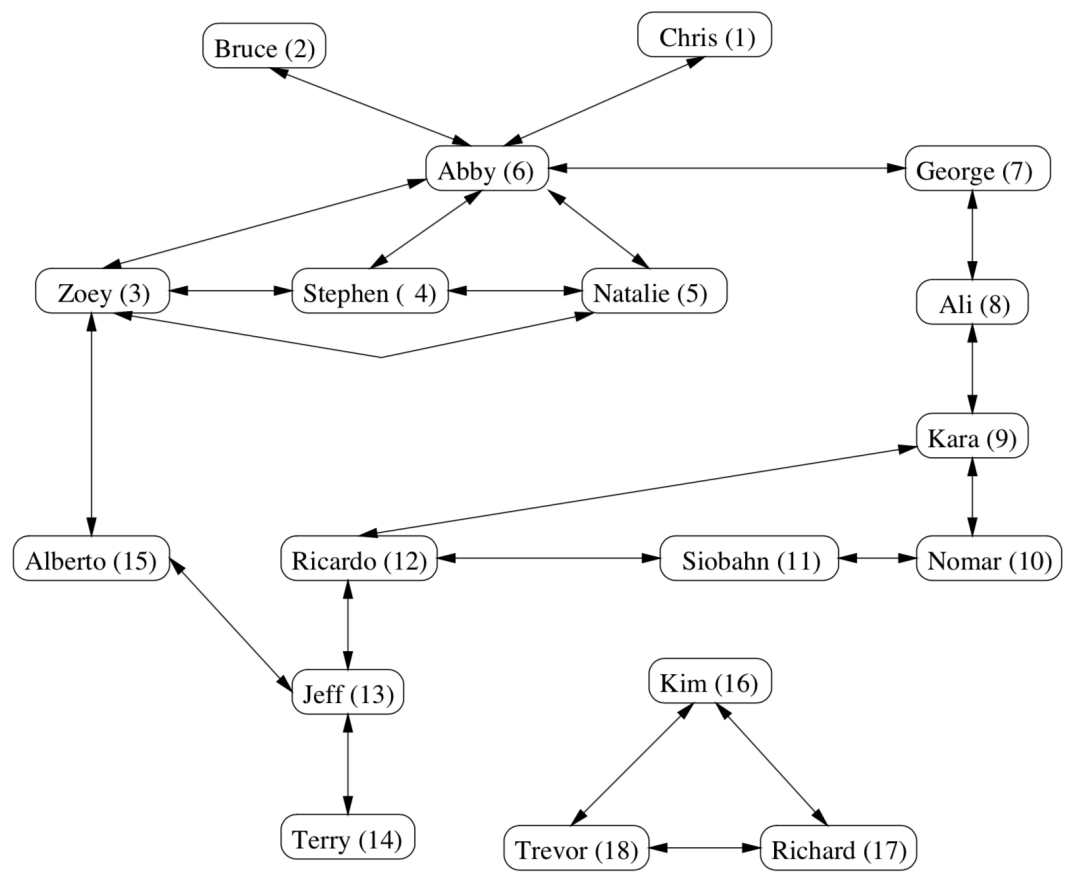
The shortest path between Abby and Alberto has two steps (Abby  $\rightarrow$  Zoey, and Zoey  $\rightarrow$  Alberto), so we say the degree of separation is 2. Additionally, Alberto would be a friend of a friend of Abby.

You can assume an initial configuration of who is friends with who as outlined in the diagram above. You will need to store these relationships in your program. These relationships can change though, and your program needs to handle these changes. In particular, friendships can begin, possibly with new people. Friendships can end. You should be able to find friends of friends and determine the degree of separation between two people.

### Input/Output Description

Your program will read in six possible commands, with the action to be performed by your program outlined below. You may assume that  $x$  and  $y$  are integers, with  $x \neq y, x \geq 1, y \geq 1, x < 50$  and  $y < 50$ . You may also assume that instructions  $(i, d, n, f, s, q)$  occur one per line and parameters (zero, one or two integers) occur one per line.

- **i x y** –make person  $x$  and person  $y$  friends. If they are already friends, no change needs to be made. If either  $x$  or  $y$  is a new person, add them.



- `d x y` –delete the friendship between person `x` and person `y`.
- `n x` –output the number of friends that person `x` has.
- `f x` –output the number of “friends of friends” that person `x` has. Notice that `x` and direct friends of `x` are not counted as “friends of friends.”
- `s x y` –output the degree of separation between `x` and `y`. If there is no path from `x` to `y`, output Not connected.
- `q` –quit the program. Sample Interaction

Sample Interaction

Input	Output	Explanation
i 20 10	(no output)	Inserting a friendship causes no output.
i 20 9	(no output)	Inserting a friendship causes no output.
n 20	2	Person 20 has two friends (10 and 9)
f 20	3	The friends of friends of 20 are 8, 11, 12
s 20 6	4	The shortest path is 20 → 9 → 8 → 7 → 6.
q	(no output)	Program quits.