The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

2011 Canadian Computing Competition: Junior Division

Sponsor:

University of Waterloo

Problem J1: Which Alien?

Problem Description

Canada Cosmos Control has received a report of another incident. They believe that an alien has illegally entered our space. A person who witnessed the appearance of the alien has come forward to describe the alien's appearance. It is your role within the CCC to determine which alien has arrived. There are only 3 alien species that we are aware of, described below:

- TroyMartian, who has at least 3 antenna and at most 4 eyes;
- VladSaturnian, who has at most 6 antenna and at least 2 eyes;
- GraemeMercurian, who has at most 2 antenna and at most 3 eyes.

Input Specification

The user will be prompted to enter two numbers. First, the user will be prompted to enter the number of antenna that the witness claimed to have seen on the alien. Second, the user will be prompted to enter the number of eyes seen on the alien.

Output Specification

The output will be the list of aliens who match the possible description given by the witness. If no aliens match the description, there is no output.

Sample Session 1 (with output shown in `text`, user input in *italics*)

```
How many antennas?
4
How many eyes?
5
VladSaturnian
```

Sample Session 2

```
How many antennas?
2
How many eyes?
3
VladSaturnian
GraemeMercurian
```

Sample Session 3

```
How many antennas?
8
How many eyes?
6
```

(Note: there is no output for Sample Session 3)

Problem J2: Who Has Seen The Wind

Problem Description

Margaret has looked at the wind floating over the prairies for a long time. After these observations, she has created a formula that will describe the altitude of a weather balloon launched from her house. In particular, her equation predicts the altitude A (in metres above the ground) at hour t after launching her balloon is:

$$A = -6t^4 + ht^3 + 2t^2 + t$$

where h is an integer value representing the humidity as a value between 0 and 100 inclusive.

Margaret is curious at what the earliest hour is (if any) that her weather balloon will hit the ground after launch, so long as it is no more than the maximum time, M, that Margaret is willing to wait. You can assume that the weather balloon touches ground when $A \leq 0$.

In order to do this, your program should use the formula to calculate the altitude when $t = 1$, $t = 2$, and so on, until the balloon touches the ground or $t = M$ is reached.

Input Specification

The input is two non-negative integers: h, the humidity factor, followed by M, the maximum number of hours Margaret will wait for the weather balloon to return to ground. You can assume $0 \leq h \leq 100$ and $0 < M < 240$.

Output Specification

The output will be one of the following possibilities:

- `The balloon does not touch ground in the given time.`
- `The balloon first touches ground at hour:`
  `T`

where T is a positive integer value representing the earliest hour when the balloon has altitude less than or equal to zero.

Sample Input 1

```
30
10
```

Output for Sample Input 1

```
The balloon first touches ground at hour:
6
```

Sample Input 2

```
70
10
```

Output for Sample Input 2

```
The balloon does not touch ground in the given time.
```

Problem J3: Sumac Sequences

Problem Description

In a sumac sequence, $t_1, t_2, .., t_m$, each term is an integer greater than or equal 0. Also, each term, starting with the third, is the difference of the preceding two terms (that is, $t_{n+2} = t_n - t_{n+1}$ for $n \geq 1$). The sequence terminates at t_m if $t_{m-1} < t_m$.

For example, if we have 120 and 71, then the sumac sequence generated is as follows:

$$120, 71, 49, 22, 27.$$

This is a sumac sequence of length 5.

Input Specification

The input will be two positive numbers t_1 and t_2, with $0 < t_2 < t_1 < 10000$.

Output Specification

The output will be the length of the sumac sequence given by the starting numbers t_1 and t_2.

Sample Input

```
120
71
```

Output for Sample Input

```
5
```

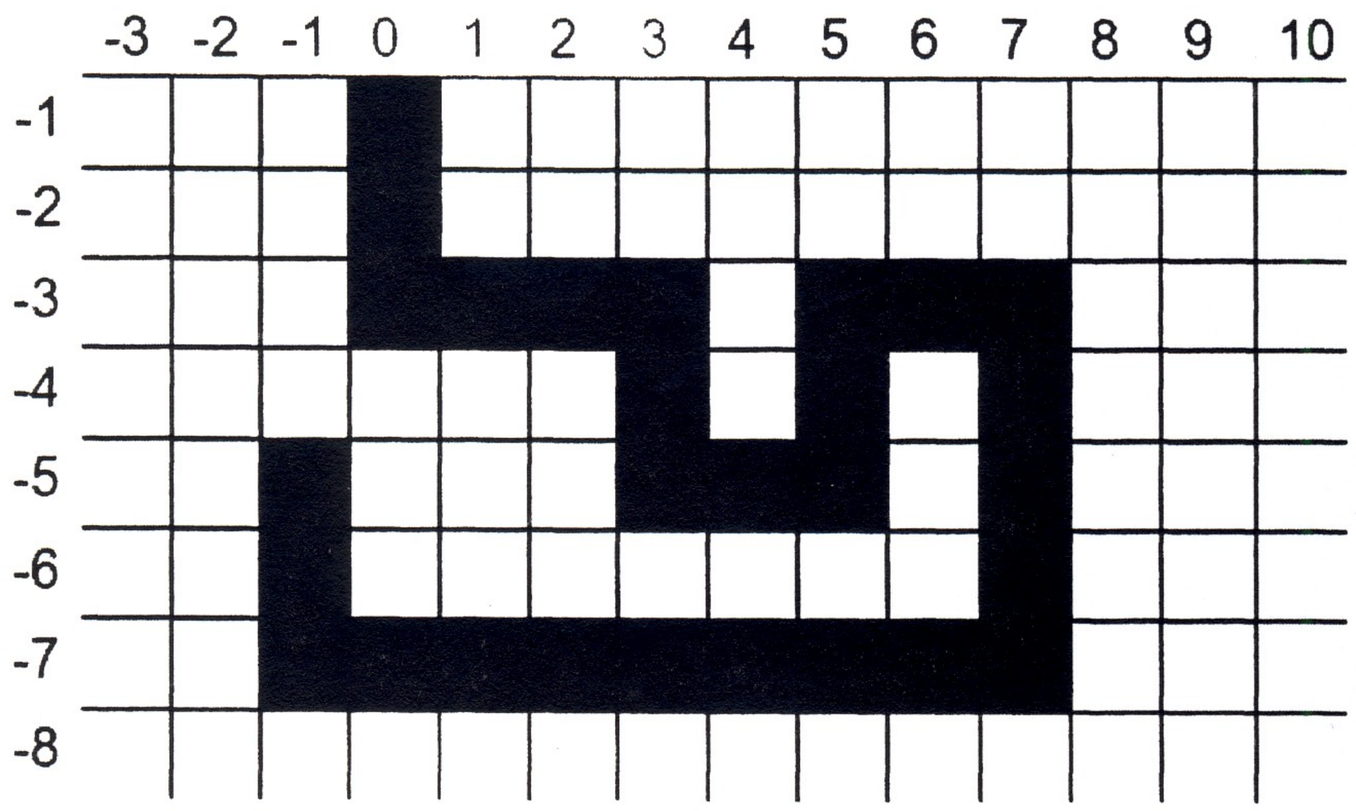
Problem J4: Boring Business

Problem Description

Boring is a type of drilling, specifically, the drilling of a tunnel, well, or hole in the earth. With some recent events, such as the Deepwater Horizon oil spill and the rescue of Chilean miners, the public became aware of the sophistication of the current boring technology. Using the technique known as geosteering, drill operators can drill wells vertically, horizontally, or even on a slant angle.

A well plan is prepared before drilling, which specifies a sequence of lines, representing a geometrical shape of the future well. However, as new information becomes available during drilling, the model can be updated and the well plan modified.

Your task is to write a program that verifies validity of a well plan by verifying that the borehole will not intersect itself. A two-dimensional well plan is used to represent a vertical cross-section of the borehole, and this well plan includes some drilling that has occurred starting at $(0, -1)$ and moving to $(-1, -5)$. You will encode in your program the current well plan shown in the figure below:



Input Specification

The input consists of a sequence of drilling command pairs. A drilling command pair begins with one of four direction indicators (`d` for down, `u` for up, `l` for left, and `r` for right) followed by a positive length. There is an additional drilling command indicated by `q` (quit) followed by any integer, which indicates the program should stop execution. You can assume that the input is such that the drill point will not:

- rise above the ground, nor
- be more than 200 units below ground, nor

- be more than 200 units to the left of the original starting point, nor
- be more than 200 units to the right of the original starting point

Output Specification

The program should continue to monitor drilling assuming that the well shown in the figure has already been made. As we can see $(-1, -5)$ is the starting position for your program. After each command, the program must output one line with the coordinates of the new position of the drill, and one of the two comments `safe`, if there has been no intersection with a previous position or `DANGER` if there has been an intersection with a previous borehole location. After detecting and reporting a self-intersection, your program must stop.

Sample Input 1

```
l 2
d 2
r 1
q 0
```

Output for Sample Input 1

```
-3 -5 safe
-3 -7 safe
-2 -7 safe
```

Sample Input 2

```
r 2
d 10
r 4
```

Output for Sample Input 2

```
1 -5 safe
1 -15 DANGER
```

Problem J5: Unfriend

Problem Description

Mark invited some people to join his social network. Some of them invited new people, who invited new people, and so on. Now there are N people in the network, numbered from 1 to N. Mark has decided to remove some people and keep others. There is one restriction: when removing a person, he will also remove the people s/he invited, and the people they invited, and so on. Mark will never remove himself, and we do not allow people to be invited by more than one person. Mark can also decide to not remove anyone.

How many different sets of people can be removed?

Input Specification:

The first line contains a single integer N ($N \leq 6$), the number of people in the network. Next are $N-1$ lines telling us who invited each person. To be precise, line i in this set ($1 \leq i \leq N-1$) contains a single integer j (with $j > i$), which indicates that person j is the person who invited person i. Person N is Mark.

Output Specification:

Output a single integer, the number of possible sets of people that can be removed.

Sample Input 1

```
3
3
3
```

Output for Sample Input 1

```
4
```

Explanation for Sample 1

The first number of the input indicates there are three people in the network. The next line tells us that Person 1 was invited by Mark, while the last line tells us that Person 2 was also invited by Mark. The sets of people that can be removed are {}, {1}, {2}, {1,2}.

Sample Input 2

```
4
3
4
4
```

Output for Sample Input 2

```
6
```

Explanation for Sample 2

There are 4 people in the network. Here is a table of who invited who:

Person inviting	Invited
1	none
2	none
3	1
4	2,3

The possible sets are {}, {1}, {2}, {1,2}, {1,3}, and {1,2,3}. Notice that the sets {3} and {2,3} are not possible, since when you remove 3, you must also remove 1.