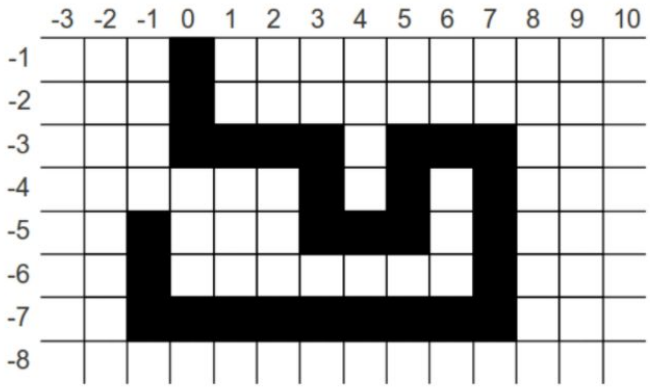
- Review of last week's homework[]
  Rotating Letters;
  Who Has Seen the Wind;
  Double Dice
  From 1987 to 2013
- Recursive Function (Important)
- 2011 Q4 Boring Business (Define variables in global scope, Two Dimensional Array)
- 2011 Q5 Unfriend (How to do the analysis,recursive function)
- 2011 Q3 Sumac Sequences (Recursive Function)
- 2020 Q5 Escape Room (HomeWork)

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.

2020 Q5

CCC '20 S2 - Escape Room

Canadian Computing Competition: 2020 Stage 1, Junior #5, Senior #2

You have to determine if it is possible to escape from a room. The room is an M-by-N grid with each position (cell) containing a positive integer. The rows are numbered $1, 2, \ldots, M$ and the columns are numbered $1, 2, \ldots, N$. We use (r, c) to refer to the cell in row r and column c.

You start in the top-left corner at $(1, 1)$ and exit from the bottom-right corner at (M, N). If you are in a cell containing the value x, then you can jump to any cell (a, b) satisfying $a \times b = x$. For example, if you are in a cell containing a 6, you can jump to cell $(2, 3)$.

Note that from a cell containing a 6, there are up to four cells you can jump to: $(2, 3)$, $(3, 2)$, $(1, 6)$, or $(6, 1)$. If the room is a 5-by-6 grid, there isn't a row 6 so only the first three jumps would be possible.

Input Specification

The first line of the input will be an integer M $(1 \leq M \leq 1\,000)$. The second line of the input will be an integer N $(1 \leq N \leq 1\,000)$. The remaining input gives the positive integers in the cells of the room with M rows and N columns. It consists of M lines where each line contains N positive integers, each less than or equal to $1\,000\,000$, separated by single spaces.

For 1 of the 15 available marks, $M = 2$ and $N = 2$.

For an additional 2 of the 15 available marks, $M = 1$.

For an additional 4 of the 15 available marks, all of the integers in the cells will be unique.

For an additional 4 of the 15 available marks, $M \leq 200$ and $N \leq 200$.

Output Specification

Output `yes` if it is possible to escape from the room. Otherwise, output `no`.

Sample Input

```
3
4
3 10 8 14
1 11 12 12
6 2 3 9
```

Copy

Output for Sample Input

```
yes
```

Copy

Explanation of Output for Sample Input

Starting in the cell at $(1, 1)$ which contains a 3, one possibility is to jump to the cell at $(1, 3)$. This cell contains an 8 so from it, you could jump to the cell at $(2, 4)$. This brings you to a cell containing 12 from which you can jump to the exit at $(3, 4)$. Note that another way to escape is to jump from the starting cell to the cell at $(3, 1)$ to the cell at $(2, 3)$ to the exit.

Notes

1. The online grader begins by testing submissions using the sample input. All other tests are skipped if the sample test is not passed. If you are only attempting the first three subtasks (the first 7 marks), then you might want to handle the specific values of the sample input as a special case.
2. For the final subtask (worth 2 marks), if you are using Java, then `Scanner` will probably take too long to read in the large amount of data. A much faster alternative is `BufferedReader`.