CCC JUNIOR LEC12

Topic: PAST EXAMS

今日课程预览

- Homework
- 2012 J5 A coin Game
- Past Exams
- 2017 J4 Favorite Times
- 2017 J5 Nailed It!
- 2014 J5 Assigning Partners
- Leet code: Two Sums
- 2013 J5 Chances of Winning
- 2011 J4 Boring Business
- 2011 J5 Unfriend

Problem J4: Favourite Times

Time limit: 1 second

Problem Description

Wendy has an LED clock radio, which is a 12-hour clock, displaying times from `12:00` to `11:59`. The hours do not have leading zeros but minutes may have leading zeros, such as `2:07` or `11:03`.

When looking at her LED clock radio, Wendy likes to spot arithmetic sequences in the digits. For example, the times `12:34` and `2:46` are some of her favourite times, since the digits form an arithmetic sequence.

A sequence of digits is an *arithmetic sequence* if each digit after the first digit is obtained by adding a constant common difference. For example, 1,2,3,4 is an arithmetic sequence with a common difference of 1, and 2,4,6 is an arithmetic sequence with a common difference of 2.

Suppose that we start looking at the clock at noon (that is, when it reads `12:00`) and watch the clock for some number of minutes. How many instances are there such that the time displayed on the clock has the property that the digits form an arithmetic sequence?

Input Specification

The input contains one integer D ($0 \leq D \leq 1\ 000\ 000\ 000$), which represents the duration that the clock is observed.

For 4 of the 15 available marks, $D \leq 10\ 000$.

Output Specification

Output the number of times that the clock displays a time where the digits form an arithmetic sequence starting from noon (`12:00`) and ending after D minutes have passed, possibly including the ending time.

Sample Input 1

```
34
```

Output for Sample Input 1

```
1
```

Explanation of Output for Sample Input 1

Between `12:00` and `12:34`, there is only the time `12:34` for which the digits form an arithmetic sequence.

Sample Input 2

```
180
```

Output for Sample Input 2

```
11
```

Explanation of Output for Sample Input 2
Between `12:00` and `3:00`, the following times form arithmetic sequences in their digits (with the difference shown:

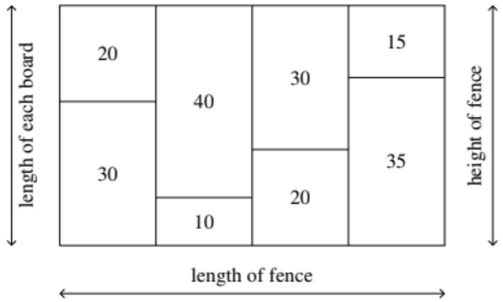
- `12:34` (difference 1),
- `1:11` (difference 0),
- `1:23` (difference 1),
- `1:35` (difference 2),
- `1:47` (difference 3),
- `1:59` (difference 4),
- `2:10` (difference -1),
- `2:22` (difference 0),
- `2:34` (difference 1),
- `2:46` (difference 2),
- `2:58` (difference 3).

Problem J5: Nailed It!

Time limit: 2 seconds

Problem Description
Tudor is a contestant in the Canadian Carpentry Challenge (CCC). To win the CCC, Tudor must demonstrate his skill at nailing wood together to make the longest fence possible using boards. To accomplish this goal, he has N pieces of wood. The i^{th} piece of wood has integer length L_i.

A *board* is made up of **exactly two** pieces of wood. The length of a board made of wood with lengths L_i and L_j is $L_i + L_j$. A *fence* consists of boards that are the same length. The *length of the fence* is the number of boards used to make it, and the *height of the fence* is the length of each board in the fence. In the example fence below, the length of the fence is 4; the height of the fence is 50; and, the length of each piece of wood is shown:



Tudor would like to make the longest fence possible. Please help him determine the maximum length of any fence he could make, and the number of different heights a fence of that maximum length could have.

Input Specification
The first line will contain the integer N ($2 \leq N \leq 1\,000\,000$).
The second line will contain N space-separated integers $L_1, L_2, \ldots, L_N$ ($1 \leq L_i \leq 2\,000$).

For 7 of the 15 available marks, $N \leq 100$.
For an additional 6 of the 15 available marks, $N \leq 1000$.
For an additional 1 of the 15 available marks, $N \leq 100\,000$.

Output Specification
Output two integers on a single line separated by a single space: the length of the longest fence and the number of different heights a longest fence could have.

Sample Input 1
```
4
```

```
1 2 3 4
```

Output for Sample Input 1

```
2 1
```

Explanation for Output for Sample Input 1

Tudor first combines the pieces of wood with lengths 1 and 4 to form a board of length 5. Then he combines the pieces of wood with lengths 2 and 3 to form another board of length 5. Finally, he combines the boards to make a fence with length 2 and height 5.

Sample Input 2

```
5
1 10 100 1000 2000
```

Output for Sample Input 2

```
1 10
```

Explanation for Output for Sample Input 2

Tudor can't make a fence longer than length 1, and there are 10 ways to make a fence with length 1 by choosing any two pieces of wood to nail together. Specifically, he may have a fence of height 11, 101, 1001, 2001, 110, 1010, 2010, 1100, 2100 and 3000.

Problem J5: Assigning Partners

Problem Description

The CEMC is organizing a workshop with an activity involving pairs of students. They decided to assign partners ahead of time. You need to determine if they did this *consistently*. That is, whenever A is a partner of B, then B is also a partner of A, and no one is a partner of themselves.

Input Specification

The input consists of three lines. The first line consists of an integer N $(1 < N \leq 30)$, which is the number of students in the class. The second line contains the first names of the N students separated by single spaces. (Names contain only uppercase or lowercase letters, and no two students have the same first name). The third line contains the same N names in some order, separated by single spaces.

The positions of the names in the last two lines indicate the assignment of partners: the ith name on the second line is the assigned partner of the ith name on the third line.

Output Specification

The output will be `good` if the two lists of names are arranged consistently, and `bad` if the arrangement of partners is not consistent.

Sample Input 1

```
4
Ada Alan Grace John
John Grace Alan Ada
```

Output for Sample Input 1

```
good
```

Explanation for Output for Sample Input 1

Ada and John are partners, and Alan and Grace are partners. This arrangement is consistent.

Sample Input 2

```
7
Rich Graeme Michelle Sandy Vlado Ron Jacob
Ron Vlado Sandy Michelle Rich Graeme Jacob
```

Output for Sample Input 2

```
bad
```

Explanation for Output for Sample Input 2

Graeme is partnered with Vlado, but Vlado is partnered with Rich. This is not consistent. It is also inconsistent because Jacob is partnered with himself.

1. Two Sum

Easy 18272 652 Add to List Share

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Output: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

Constraints:

- $2 <= nums.length <= 10^3$
- $-10^9 <= nums[i] <= 10^9$
- $-10^9 <= target <= 10^9$
- **Only one valid answer exists.**

Problem J5: Chances of winning

Problem Description
You want to determine the chances that your favourite team will be the champion of a small tournament.

There are exactly four teams. At the end of the tournament, a total of six games will have been played with each team playing every other team exactly once. For each game, either one team wins (and the other loses), or the game ends in a tie. If the game does not end in a tie, the winning team is awarded three points and the losing team is awarded zero points. If the game ends in a tie, each team is awarded one point.

Your favourite team will only be the champion if it ends the tournament with strictly more total points than every other team (i.e., a tie for first place is not good enough for your favourite team).

The tournament is not over yet but you know the scores of every game that has already been played. You want to consider all possible ways points could be awarded in the remaining games that have not yet been played and determine in how many of these cases your favourite team will be the tournament champion.

Input Specification
The first line of input will contain an integer T which is your favourite team ($1 \leq T \leq 4$).
The second line will contain an integer G, the number of games already played ($0 \leq G \leq 5$).
The next G lines will give the results of games that have already been played. Each of these lines will consist of four integers A, B, S_A, S_B separated by single spaces where $1 \leq A < B \leq 4$, and $S_A, S_B \geq 0$. This corresponds to a game between team A (which had score S_A) and team B (which had score S_B) where team A won if $S_A > S_B$, team B won if $S_A < S_B$ and the game ended in a tie if $S_A = S_B$. The pairs A and B on the input lines are distinct, since no pair of teams plays twice.

Output Specification
The output will consist of a single integer which is the number of times that team T is the champion over all possible awarding of points in the remaining games in the tournament.

Sample Input 1
```
3
3
1 3 7 5
3 4 0 8
2 4 2 2
```

Output for Sample Input 1
```
0
```

Explanation of Output for Sample Input 1
Team 3 has lost two of its three games, and team 4 has tied one and won one, which gives 4 points to team 4. Even if team 3 wins its final game, it cannot have more points than team 4, and thus, will not be champion.

Sample Input 2

```
3
4
1 3 5 7
3 4 8 0
2 4 2 2
1 2 5 5
```

Output for Sample Input 2

```
9
```

Explanation of Output for Sample Input 2
After these games, we know the following:

Team	Points
1	1
2	2
3	6
4	1

There are two remaining games (team 3 plays team 2; team 1 plays team 4). Teams 1, 2 or 4 cannot achieve 6 points, since even if they win their final games, their final point totals will be 4, 5 and 4 respectively. Thus, out of the 9 possible outcomes (2 matches with 3 different possible results per match), team 3 will be the champion in all 9 outcomes.

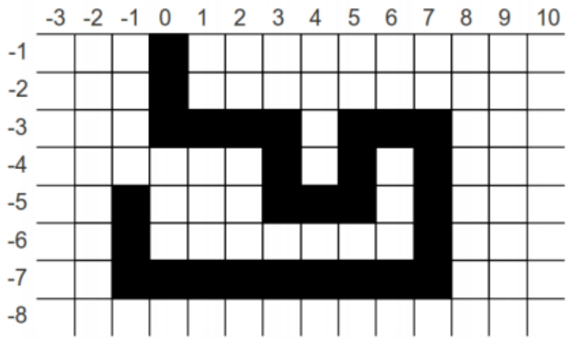
Problem J4: Boring Business

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.