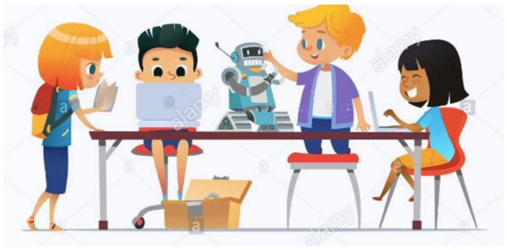
Assignment#5



(Advanced Python)

Don't be afraid to look at resources online or peek at the solutions (if you can find them on the internet).

Problem#1

Activity Buddies

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.

Problem#2

(Drilling)

Hint: 2D array (or 2D lists) should be created.

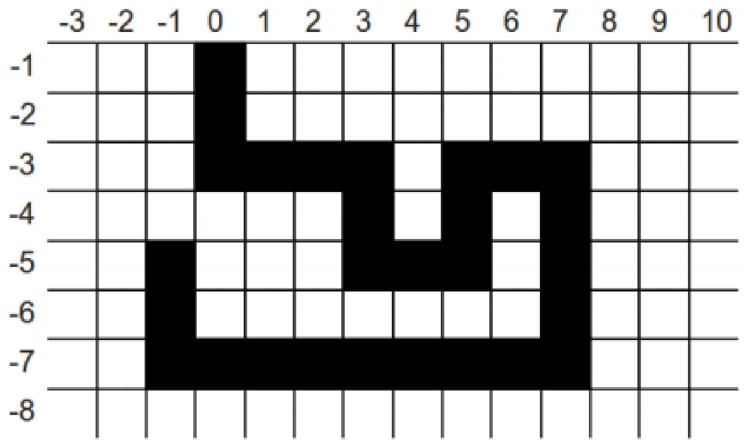
Problem J4: Boring Business

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
```