Answer three (3) of the following problems using the Java, C#, C/C++, Python, or PHP language. Please provide source code to each project separated by folders, named for the respective question that is being answered (q1, q2, q3, etc.), in a single zipped file within 48 hours of receiving this test.

Please note, we consider elegant design and clean coding style equally as important as a correct solution.

Each program should accept a file, the full path of which should be a parameter passed into your program, as input. The output should be sent to standard out. Each problem below includes sample input and sample output. Please conform your input and output to the samples provided. Also note that the "***" characters are for delineation only and should not be included as input or as output.

Example:

```
root@localhost> java FindSource /home/root/devtest/q1/example.txt
3.0 4.0
root@localhost>
```

#1 - Finding the source

In a field, we have hidden an object with a radio transmitter. However, a battery failure has made it so that the transmitter emits pulses only once every 6 hours. You have already taken measurements at two points and, from the strength of the signal, determined the distance that the object must be from those points. Your job now is to write a program to find all of the locations, if any, where the object might be. You only have the two datapoints you have already taken, and you can't wait to take any more.

Your program will be given an input file that contains two data points that look like the following, each on a separate line.

<x position> <y position> <distance from object>

For example, consider the following input:

```
***
6.0 8.0 5.0
0.0 0.0 5.0
***
```

Then the output of your program should be:

```
***
3.0 4.0
***
```

which indicates that the object must reside at coordinates x = 3, y = 4.

# 2 - Justifying text

Suppose you are programming for a really old-school internet newspaper, one that is so old-school that they only have monospace fonts. However, the editor really wants to see the monospace text justified to fit into the column space he has at a particular time. Your job is to write a program that will take as input the width of the column in characters and the entire text to be formatted, and return the same text except justified to fit into the column.

For example, consider the following input:

```
***
20
The quick brown fox jumps over the lazy dog.
***
```

Then the output of your program should be:

```
***
The quick  brown fox
jumps over  the lazy
dog.
***
```

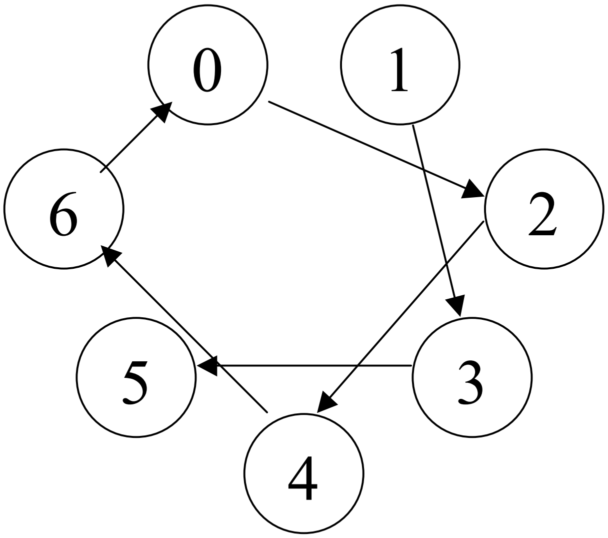
As you can tell from this example, the last line does not have to be justified.

#3 - Following the pointers

You are given a puzzle which contains N locations. From each location, you can travel to a maximum of one other location on the puzzle or to a dead end. Your job is to write a program to count how many cycles exist in the puzzle. To clarify, a cycle exists if you can start at a given location and, following the available moves from that location, eventually end up back where you started.

The input your program will be given will be a simplification of the puzzle. The first line of input will indicate how many locations there are in the puzzle, call it N as above. The next N lines show connections between locations. In particular, if line I contains number J, then from location I you can get to location J and only location J. Any location that is a dead end will contain the integer -1. Note that the locations are numbered from 0 to N-1.

Example puzzle:



For example, consider the following input:

```
***
7
2
3
4
5
6
-1
0
***
```

Then the output of your program should be:

```
***
```

1

***

#4 – String Manipulation

Write a function to compact a string **in place**:

A. strip whitespace from the string.
B. remove duplicate characters if they are next to each other

For example, consider the following input:

```
***
abb cddpddef gh
***
```

Then the output of your program should be:

```
***
abcdpdefgh
***
```

#5 - Spiral printing

Write a function to print a 2-D array (n x m) in spiral order (clockwise).

For example, consider the following input:

```
***
1 2 3
4 5 6
7 8 9
***
```

Then the output of your program should be:

```
***
1 2 3 6 9 8 7 4 5
***
```

# 6 – Sub-Trees

Write a function to check if binary tree 1 is a subtree of another binary tree 2. Tree 1 is a subtree of Tree 2 if and only if we can find a subtree in Tree 2 that has the same structure as Tree 1 and each node has the same value. Input for this problem will use a breadth-first heap style representation.

For example:
4,2,5

Represents this tree:

```
  4
 / \
2   5
```

Which is a subtree of:
1,5,4,,3,2,5,,,,,,,0,8

```
  1
 / \
5   4
 \  / \
 3 2   5
       /\
      0  8
```

But it's not a subtree of:
1,5,4,,3,2,9,,,,,,,0,8

```
  1
 / \
5   4
 \  / \
 3 2   9
       /\
      0  8
```

For example, consider the following input:

```
***
1,5,4,,3,2,5,,,,,,,0,8
4,2,5
***
```

Then the output of your program should be:

```
***
Yes
***
```

Hint:

How to create a binary tree? Visit: http://cslibrary.stanford.edu/110/BinaryTrees.html

Sample struct and function in C:

```
typedef struct {
   int val;
   Node *left;
   Node *right;
}Node;

Node *subtree(Node *t1, Node *t2);
```