

ND Programming Contest, Spring 2010

ND ACM Computer Club

Saturday April 10, 2010

#1: The Easiest Problem in the World.

The Problem

You are currently about to embark upon the computer programming contest challenge. Unfortunately, the people running the contest goofed on the really easy problem and encrypted it such that it is almost impossible to solve. Based upon a top secret government project, however, you think you know how to break the code. The code uses modulo addition, where two numbers are added together as follows. $3+2=3$
Translated, this means: 3

The impressive part about this encryption is that it sometimes results in the addition of two positive numbers being equal to zero. This usually throws off any hackers who try to break the code. Your mission, should you choose to accept it, will be to decode a list of numbers.

Sample Input

Input will consist of a series of numbers to be added together, with each number separated by a space. Each new line will correspond to a new set of numbers. Following the final data set will be a single blank line.

```
1 2
2 3
3 4
5 1
6 2
7 3
10 10
```

Sample Output

Output will consist of one number per line corresponding to the arithmetic answer.

```
1
3
4
1
2
4
0
```

#2: Economic Encounters

The Problem

Due to the recent problems with the economy, you have been hired as an accountant for a multi-billion dollar grocery chain. In the past, they did not have to worry about loose change, but that is no longer the case. They need for you to create a program that will return the correct amount of change, using the least amount of coins possible. Be careful! Recent reports show that the economy has been recovering, but if this problem is solved incorrectly, the consequences could be devastating. The entire grocery business could collapse and people might go hungry.

Sample Input

The input will begin with the desired value of change to be given, one value per line. Input will end with a blank line.

```
3
1
5
20
75
```

Sample Output

The output will consist of a series of numbers corresponding to each value of a coin from 25..10..5..1, as shown below.

```
0 0 0 3
0 0 0 1
0 0 1 0
0 2 0 0
3 0 0 0
```

#3: Word Search Wizardry

The Problem

The world word search champion has recently challenged you to a life-or-death game of word search. You, recognizing you can't possibly win by your own meager word search capabilities, decide that it's time to take advantage of a loop hole that allows you to use a computer and put your programming knowledge to the test. Your opponent has not revealed the puzzle size or the number of words yet, but you do know from an outside source that there will not be any diagonal words. For this reason your program should search through the grid both horizontally and vertically for all of the words in the list and output their locations in the order they were presented.

Sample Input

Input will consist of the grid size followed by each separate row of letters. On the next line will be the word count and then finally each word to be found on a separate line.

```
5 5
a b c d n
s a z n o
y l i a t
k l u q r
u d a m e
3
notre
dame
qand
```

Sample Output

The output should be formatted as such: [first letter row] [first letter column] [last letter row] [last letter column]. Be sure to output the

```
1 5 5 5
5 5 5 2
4 4 2 4
```

#4: Bullying the Boss

The Problem

Your boss, looking for a way to cut down on the amount of paper being used, has asked that you develop a program that will remove the spaces in a line and convert every uppercase letter to a lowercase letter. Feeling spiteful and not particularly attached to your job you decide the program should reverse the order of each line as well. Only spaces are to be removed so all other characters must be left intact.

Sample Input

Take in the number of strings that should be reversed followed by the strings themselves on separate lines.

```
3
McRib is B A C K
Live not on eViL madam Live not on eViL
198 @.23 23#5 4 5 874
```

Sample Output

For the output you should return the newly formatted string with all whitespaces removed, letters changed to lowercase, and order reversed.

```
kcabsibircm
livenotonevilmadamlivenotonevil
478545#3232.@891
```

#5: Sprinkler Dodging

The Problem

While trying to get to class one morning you realize that all of the sprinklers on campus are turned on. You are carrying a paper that you have to turn in when you get to class so you can't afford to get wet. Your job is to try to find a path from your dorm to your class that stays on campus and doesn't come within range of the sprinklers.

Campus will be represented by a 20 by 20 box, with your dorm located at (0,0) and your classroom at (20,20).

Sample Input

Input will begin with the number of test cases. Each test case will start with the number of sprinklers. Each sprinkler will be represented by 3 integers the x coordinate, the y coordinate, and the radius it shoots water.

```
3
1
19 19 1
5
20 0 5
15 5 5
10 10 3
5 15 5
0 15 5
2
10 10 5
5 15 5
```

Sample Output

For each test case, your program should output whether or not there is a path from your dorm to class without getting wet.

```
no
yes
yes
```

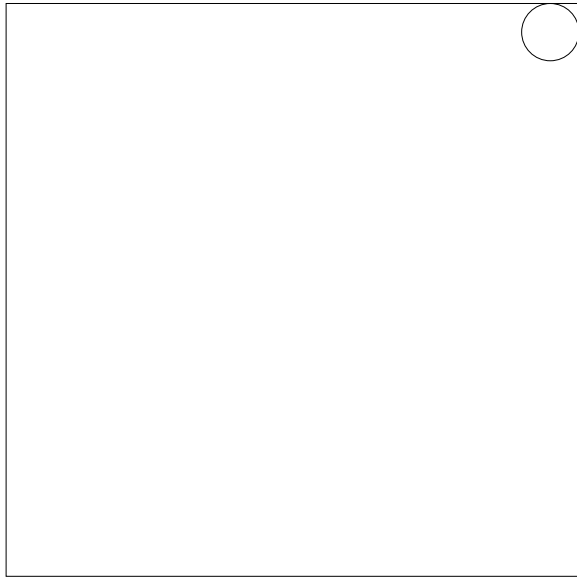


Figure 1: Sample input 1 sprinkler system

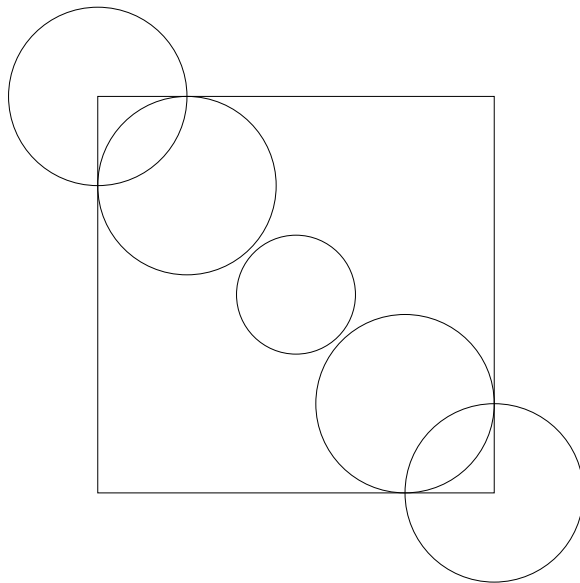


Figure 2: Sample input 2 sprinkler system

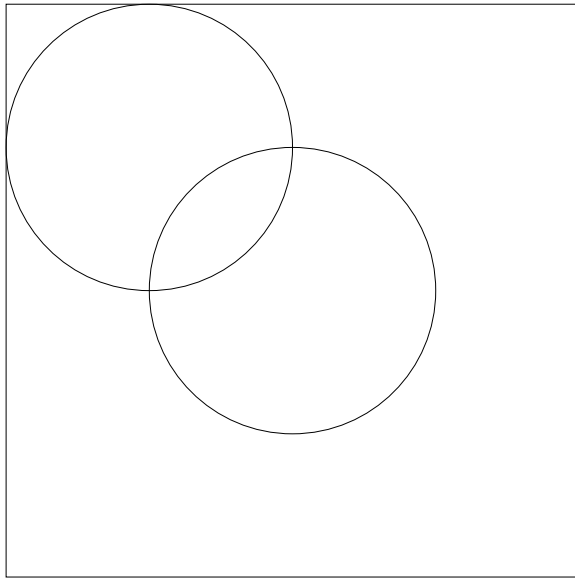


Figure 3: Sample input 3 sprinkler system