COMPUTER SCIENCES

George Fox University H.S. Programming Contest Practice Problems 2022

General Notes

1. Do the problems in any order you like. They do not have to be done in order
(hint: the easiest problem may not be the first problem)

2. Scoring: The team who solves the most problems in the least amount of time with the least submissions wins. Each wrong submission will receive a 20 min time penalty that will only be added to the time score once the problem has been successfully solved. Time is calculated for each problem as the total time from the start of the contest to the time it was solved.

3. There is no extraneous input. All input is exactly as specified in the problem. Integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Do not welcome the user. Do not prompt for input. Follow the form exactly as given in the problem.
(hint: spaces? No spaces? What does spec say!)

5. All solutions must be a single source code file.

This page intentionally left blank

1. PPrime

Prime palindromes are prime numbers that are the same number forwards and backwards. Here are some examples:
11, 101, 131, 151, 727….
In this program, determine whether a number is a palindromic prime, with at least 2 digits, and less than 10 digits.

Input
The first line (N) consists of the number of data sets. Each data set contains one number.

Output
For each number, print out "yes" or "no."

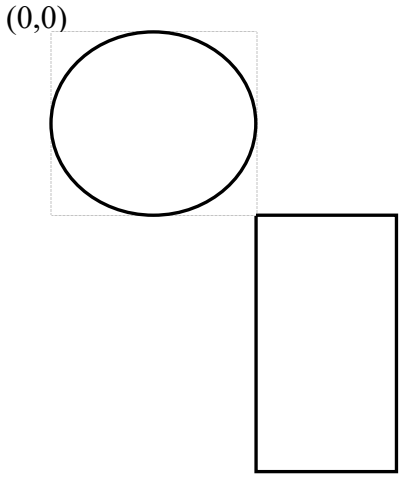
Example Input:
```
6
5
131
727
152
83
30203
```

Output to screen:
```
no
yes
yes
no
no
yes
```

This page intentionally left blank

2. Draw

Some CS classes use GUI (graphical user interface) examples to learn about methods, variables, and OOP concepts. So for this program, you will "pretend" to use the drawRect() and drawOval() methods along with mousePressed(), getX() and getY() to get the coordinates of a mouse event. Given the mouse event coordinates, the oval coordinates and the rectangle coordinates, use if statements to determine if the click was inside the circle, the rectangle, or neither (or on the final pixel itself, so be inclusive). The upper left corner of a window (or panel) is (0,0) with x increasing to the right, and y increasing downward.



The method drawOval(int x, int y, int wid, int len) requires the upper left coordinate x and y and the width (horizontal) and length (vertical) of the outline of the oval. You need to calculate the center and radius of the circle!
The method drawRect(int x, int y, int wid, int len) requires the upper left coordinate x and y and the width and length.

For this program, the window is 600 by 600 pixels. The circle and rectangle will not overlap, so the mouse click can only be in one object or the other or in neither.

NOTE: The judge data will have different coordinates for the rectangle and circle, so don't hard-code those values.

Input
The first line contains the coordinates for the oval/circle: x, y, wid (wid=len).
The second line contains the coordinates for the rectangle: x, y, wid, len.
The third line contains a single integer N which indicates how many mouse clicks were made.
The following N input lines contain the x and y coordinates of the mouse click.

Output
Print out "circle", "rectangle", or "neither" for each mouse click.

(Continued on next page…)

(2. Draw continued)

Example Input:

```
10 10 50
100 200 50 300
6
5 5
40 40
101 201
150 502
50 50
35 61
```

Output to screen:

```
neither
circle
rectangle
neither
circle
neither
```

Explanation of example test cases above:

The circle has a center at (35,35) with radius 25.
The rectangle is upper left (100, 200) and lower right (150, 500).
In the 1st data set, neither—above and left of the circle.
In the 2nd data set, circle.
In the 3rd data set, rectangle – upper left hand corner.
In the 4th data set, neither – below rectangle.
In the 5th data set, circle – lower right in the circle.
In the 6th data set, neither – below the circle.

3. Fence

I built a wooden fence for my yard 3 years ago. To protect the wood, I use a coating called Wood Defender every 2 years. The directions on the container says that 1 gallon will cover 150 ft^2. So in designing a website for this product, an online calculator would need the input for the length and height of the fence. Calculate how many gallons someone would need to buy for the size of the fence. You will need to coat both sides of the fence.

Input
The first line (N) consists of the number of data sets. Each data set contains two integers, the length and height of the fence (in feet).

Output
Print out an integer, the number of gallons needed.

Example Input:
```
3
7 60
6 50
8 120
```

Output to screen:
```
6
4
13
```

Explanation of example test cases above:
In the 1st data set, 7*60*2 = 840 ft^2 and 840/150 = 5.6 (use 6 gallons).
In the 2nd data set, 6*50*2 = 600 ft^2 and 600/150 = 4.0 (use 4 gallons).
In the 3rd data set, 8*120*2 = 1920 ft^2 and 1920/150 = 12.8 (use 13 gallons).

This page intentionally left blank

4. Vowel

What if you compared the vowels in words of the same length at the same positions? You could see if the letters in both were vowels, or not, or if only one letter were a vowel and the other a consonant. Sounds like bitwise operators. Vowels are considered to be [aeiou] for this problem.

Given two words of equal length, convert the word to a binary number for the AND (&), OR(|), XOR(^) . Here is an example with the words, happy and shops:

```
happy
shops
00000 = 0 (AND – no chars are both vowels)
01100 = 12(OR – 2nd and 3rd chars are vowels)
01100 = 12(XOR is the same as OR for these words)

happy
caper
01000 = 8 (AND – 2nd char are both vowels)
01010 = 10(OR – 2nd and 4th chars are vowels)
00010 = 2 (XOR – 4th chars have only 1 vowel, but 2nd are both, not
exclusive)
```

Input
The first line (N) consists of the number of data sets used in this program. Each N following lines contains two words separated by a space. Each word is less than 32 characters and at least 1 character.

Output
Print out the bitwise &, |, and ^ value for each word pair on one line separated by a space.

Example Input:

```
4
happy shops
happy caper
somebody everyone
at in
```

Output to screen:

```
0 12 12
8 10 2
4 245 241
2 2 0
```

This page intentionally left blank

5. Yoda

I like Star Wars, as do many others. Probably some of you do too if you're into CompSci. If you haven't seen the 7 episodes of the Star Wars saga, that is your homework next weekend. Seriously. You need to see these movies.

Yoda has been one of the iconic characters in the sci-fi realm for many years (since 1981 when The Empire Strikes Back debuted in theaters). Frank Oz was the voice of Yoda. His unique speech pattern was part of why we all love Yoda. In trying to find an algorithm that would describe his speech patterns, here's what I determined. For many of his sentences, it seems that the first two words are put at the end of a sentence. For example, you and I might say, "This is my home." But Yoda would say, "My home this is." Convert a regular English sentence into Yoda-ese.

Input
The first line (N) consists of the number of data sets used in this program. Each N following lines contain a sentence, lowercase with no punctuation. Each sentence will have at least 3 words.

Output
Print out each converted sentence.

Example Input:
```
4
this is my yoda shirt
i am happy to see you
this is my home
you are reckless
```

Output to screen:
```
my yoda shirt this is
happy to see you i am
my home this is
reckless you are
```

This page intentionally left blank

6. Tic Tac Toe

It's Summer vacation for Alex and Ben, which means lots of sitting around at home while their parents are at work. Luckily you've been hired as their sitter, and you've brought tons of games to ease their boredom. You've decided to start them off with an easy game of tic tac toe and ease into more complicated areas of game theory later. To help them grasp the game you've decided to write a program to simulate a game as well given a valid set of moves.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `m` denoting how many moves take place in each game. The following `m` lines will each consist of two integers `r` and `c`, representing the row and column of each move, with X and O alternating turns marking one spot. X's will always go first, and every set of moves will be legal.

Output

For each test case, recreate the board after the sequence of moves, and then report either of the following outcomes accordingly. Leave a blank line between test cases.

- `X wins!`
- `O wins!`
- `Tie Game!`
- `Incomplete`

Example Input

```
3
7
0 0
1 2
0 2
0 1
1 0
1 1
2 0
1
1 1
9
0 0
1 1
0 2
0 1
2 1
1 0
1 2
2 2
2 0
```

Example Output to Screen

```
X|O|X
-----
X|O|O
-----
X| |
X wins!

 | |
-----
 |X|
-----
 | |
Incomplete

X|O|X
-----
O|O|X
-----
X|X|O
Tie Game!
```

This page intentionally left blank

7. Rock Paper Scissors

You're now introducing Alex and Ben to Rock Paper Scissors, and you've been able to work out a pattern to predict all of their moves. Every time they lose a game, they will switch to whatever would beat what they lost to, without fail. For instance, if they picked rock and lost to paper, their next move would always be scissors. In any other case, they tend to just pick the same thing over and over again, and they always start on rock. Making use of this pattern, Write a program to calculate how many games you will win lose and tie against Alex and Ben given a series of move choices.

Input
The first line will contain a single integer `n` that indicates the number of lines that follow. The following `n` lines each contain a sequence of `R`, `P`, and `S` representing you choosing Rock, Paper, and Scissors respectively.

Output
Output the number of wins, losses, and ties in the following format:
```
Wins: X
Losses: Y
Ties: Z
```
And leave a blank line between test cases

Example Input
```
3
RRRPRP
PRSPRSPRS
SSSSSS
```

Example Output to Screen
```
Wins: 2
Losses: 0
Ties: 4

Wins: 9
Losses: 0
Ties: 0

Wins: 0
Losses: 6
Ties: 0
```

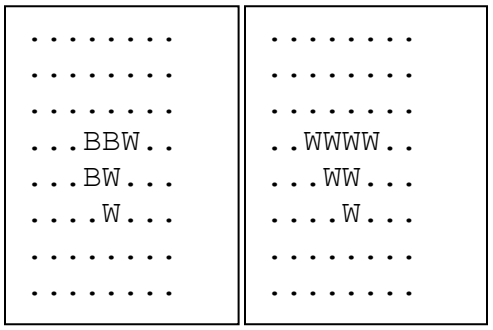
This page intentionally left blank

8. Othello

Othello is a game played with black and white disks on an 8x8 grid. Two players take turns placing a disk of their color and attempting to "sandwich" as many of the opponent's disks as possible between two of theirs, and then replacing those with their own. For example, consider this move:

By putting a white tile at spot (3,2), the black tiles at (3,3), (3,4), And (4,3) are all sandwiched between (3,2) and another white disk, so the resulting board would have all of these flipped to white.

Given a board configuration and a potential move, determine what The game board would look like after making that move.



Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with 8 lines each consisting of 8 characters, with '`B`', '`W`', and '`.`' Representing black, white, and empty tiles respectively. The next line will consist of two integers `r` and `c` and either a `W` or `B`, representing the row, column, and disk color of the potential move.

Output

Output the Othello grid after the given move takes place. If the potential move would not result in any disks being flipped, print "`Invalid Move`" instead. Print a blank line between test cases.

Example Input

```
2
........
........
........
...BBW..
...BW...
....W...
........
........
3 2 W
........
........
.....B..
...WWWWW
...B..B.
........
........
........
3 2 B
```

Example Output to Screen

```
........
........
........
..WWWW..
...WW...
....W...
........
........

Invalid Move
```

This page intentionally left blank

9. Rooks

You've now broken out the chess board, but Alex and Ben are having a bit of trouble. At their request, you've removed everything except for the rooks. But this got the three of you wondering; given an unlimited amount of rooks, and any square board size, how many different ways can you arrange the maximum amount of rooks without any threatening each other?

Refresher
Rooks can only move in straight lines. A rook is considered "threatened" if it shares a row or column with another rook.

Input
The first line will contain a single integer `n` that indicates the number of data sets that follow. The next `n` lines contain an integer `m` representing the size of the chess board.

Constraints
1<=m<=20

Output
For each test case, output the number of ways to arrange the maximum amount of rooks without any being threatened. This answer could be very large, so be sure to use data types accordingly.

Example Input
```
2
2
4
```

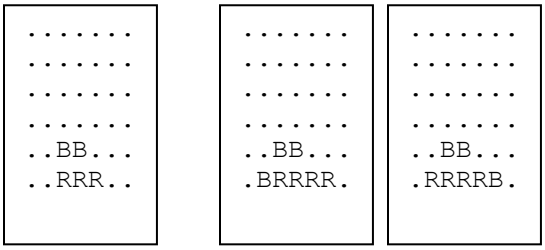
Example Output to Screen
```
2
24
```

This page intentionally left blank

10. Doubled!

The Game of Connect Four consists of two players taking turns dropping checkers of their color into a 6x7 frame, attempting to line up four of their checkers before their opponent. Sometimes however, this game becomes futile. Skilled players can "double" their opponents, creating a game position where their opponent will lose on the next turn regardless of their move. Consider the following game:

No matter where the black drops their tile, red is still capable of winning during the next turn, as shown in the two diagrams to the right.



Given a connect four game, determine whether the current player has been doubled.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single character '`B`' or '`R`', indicating which player's turn it is. The next 6 lines of each data set will each contain one line of the Connect Four grid, '`B`' representing black checkers, '`R`' being white, and '`.`' being empty spaces.

Output

For each test case, if the current player has been doubled, print "`Doubled!`".
Otherwise, print "`There's hope`".

Example Input

```
3
B
.......
.......
.......
.......
..BB...
..RRR..
R
.......
.......
.......
.......
..BB...
..RRR..
B
.......
.......
...B.R.
...BRB.
...RRR.
...RBB.
```

Example Output to Screen

```
Doubled!
There's hope
Doubled!
```

This page intentionally left blank