



## Question - 1

### Sample Question

---

This is a sample question that helps you understand how input-output works on HackerRank. You do not need to write any code, but please read the question and the answer (code) below for a better understanding, before you start solving the remaining questions.

#### Sample Question

Given two positive integers *a* and *b*, your task is to complete the function **sumNum(int a, int b)** to return their sum.

#### Sample Input

2  
3

#### Sample Output

5

## Question - 2

### The Chessboard Pattern Problem

---

Write a function **chessBoard()** that returns a chessboard pattern ("B" for black squares, "W" for white squares). The function takes a number *N* as input, and generates the corresponding board. *N* will always be  $> 0$ .

The below output comes from an input parameter of 5. The top left square will always be white. There will be *N* lines of output, each line corresponding to one row of the board. All output should be printed to stdout.

#### **Sample Input:**

5 // The size of the chess board

#### **Sample output:**

W B W B W  
B W B W B  
W B W B W  
B W B W B  
W B W B W

## Question - 3

### Number Permutations

---

Your challenge is to write a function **numPermutation()** that takes in a number *N* and returns an array of all possible permutations of the

number given . For example, if the input number is 123, the output should be:

123  
132  
213  
231  
312  
321

Please Note: It is invalid to write the number 1 as 001.

## Question - 4

### Letter Sudoku

---

Your job is to validate Letter Sudoku solutions. While ordinary Sudoku uses the numbers 0..9, Letter Sudoku uses 9 distinct letters from the alphabet. For each board, write a function **sudoku()** that takes in a string and corresponding grid, and returns a string: **"Valid"** or **"Invalid"**.

The first line of input consists of the 9 letters that form the puzzle without any spaces between the letters. The Letter Sudoku board is then input as 9 lines of 9 letters each, each letter separated from the other by a space character. A board is valid, if and only if, in every row and column, and in every 3x3 sub-board, each letter appears exactly once. It is super-important to adhere to the case in the output string. A sample Valid input is shown below:

```
MONASTERY // This is the first line of input consisting of the 9 letters
that form the puzzle
9 // denotes the number of columns in the sudoku
9 // denotes the number of rows in the sudoku
R M E T O Y N A S // These 9 lines are the Sudoku
board
N Y O S M A R E T
T S A N R E Y O M
E R M O Y S A T N
S A Y R T N O M E
O N T E A M S Y R
M O N A S T E R Y
A T S Y E R M N O
Y E R M N O T S A
```

## Question - 5

### Number Sequence Puzzle

---

What is the next number in this sequence: 1, 11, 21, 1211, 111221,....?  
Before you tear your hair out, here is the solution: In general, each number in the sequence is formed by "reading" the previous number as a "string of digits" – for each run of  $n$  instances of digit  $d$ , append  $n$  followed by digit  $d$  to the output string. For example, 1211 consists of one 1, one 2, and two 1s. Therefore, the next number in the sequence is 111221. Simple, isn't it?

Your program should take a number as input in function ***nextNumber()*** and return the next number in the sequence based on the above logic.

**Sample input:**

225

**Sample output:**

2215

## Question - 6

### Just a fraction

---

Your program input will be a fraction. A fraction consists of two integers, the numerator and the denominator.

The input is provided as 2 numbers - num and den - in the function ***justFraction()*** . Return a string as mentioned below.

Compute the reduced form of the fraction, defined as follows:  $p/q$  is reduced if and only if  $\gcd(p, q) = 1$  and  $q > 0$ . If the denominator given is 0, the function should return "Invalid"; otherwise, the function should return a string of the reduced numerator, denominator, and the decimal form rounded to three decimal places, each separated by one space. You may assume that  $-2^{31} < p, q < 2^{31}$ . Note that either the numerator or denominator could be negative. (GCD – Greatest Common Divisor)

**Sample Input**

2  
4

**Sample Output**

1 2 0.500