Perse Coding Team Challenge

Round 2 Sample Questions

There are 12 questions with 60 minutes allowed.

In this sample paper, there are different numbers of questions in each level. In the real competition, there are three questions at each of the four levels.

Each question is worth ten points.

Practice submissions may be entered and tested here:

https://www.hackerrank.com/perse-coding-cup-practice-material

Level 1 Questions

Q1) Your program will be given 3 lines of input x, y and z to specify three numbers.

- If the three numbers are all in strictly ascending order, your program must output UP.
- If the three numbers are all in strictly descending order, your program must output DOWN.
- If the three numbers are in any other order, your program must output WOBBLY.

Sample input: **Sample output:**

-1.4   UP

1.6

2.8

Q2) Your program will be given a single word w and a positive whole number n as input.

- If the word has n letters, your program must print MATCH
- If the word has more than n letters, your program must print MORE
- If the word has fewer than n letters, your program must print FEWER

Sample input: **Sample output:**

horse   MORE

4

Q3) Your program will be given a positive number n as input (integer or decimal). Print the first power of 2 that is larger than that number. For example, if num is 6 then you should print 8 because 2 cubed is 8 whereas 2 squared is only 4 and so not big enough.

Sample input: **Sample output:**

6   8

Q4) Your program should accept two lines of input. The first will be a word x (max 100 characters without spaces) and the second will be a positive whole number y (max 100). The program must print a single line of output which is the <u>reverse</u> of x repeated without spaces y times.

Sample input: **Sample output:**

Ho   oHoHoH

3

Q1) A single line of six space separated positive whole numbers will be entered. Output the sum of these numbers once having discounted any 13s (which are very unlucky) and also any number that comes immediately after a 13.

Sample input:	Sample output:
6 13 4 5 13 2	11

Q2) A single line of five space separated lowercase letters will be entered. Output the number of alphabet positions between the alphabetically least and the alphabetically greatest letter in the list. For example if the letters b and h are these least and greatest alphabetically then the output will be 6 because you have to count on six letters starting from b to reach h.

Sample input:	Sample output:
d g b h b	6

Q3) The well-known higher-lower game is played by one person choosing a secret whole number between 1 and 100 inclusive and the other person trying to guess it. After each guess the first person says either 'correct' or 'higher' or 'lower' and the guesser continues with more guesses as needed until they have correctly identified the number. The score is the number of guesses needed including the first guess and the final correct guess.

Assuming that the guesser always guesses the half-way number in the possibilities, rounding down if there are two middle numbers, then output the score achieved for the given input which represents a starting (secret) number.

For example, if the secret number is 29 then the guessing strategy would be 50,25,37,31,28,29 and so six guesses are required in total.

Sample input:	Sample output:
29	6

Level 3 Questions

Q1) A set of characters is a palindrome if it reads the same left to right as it does right to left e.g. abba, aba and cztzc are all palindromes. If we start with an input of a set of characters consisting of lower-case letters a-z (no punctuation, digits or spaces) then either it is a palindrome or it can be made one by continually removing single characters using the following simple rules:

- If a palindrome can be reached by removing a single character then remove that one. If there is more than one way to do this then remove the left-most.
- Otherwise simply remove the first character.

Note that this will not find the longest palindrome within the text necessarily but it will by default always eventually reach a palindrome because every single letter is one.

Output the first palindrome reached using this method on the given single line of input. For example if the initial input is acddcf then a palindrome cannot be reached by removing a single character and so the first character is removed leaving cddcf. However now a palindrome can be reached by removing the final character (the f) and so this is removed and a palindrome has now been reached and so cddc is printed out.

Sample input: **Sample output:**

acddcf   cddc

Q2) A number sequence can be produced by describing the previous number, starting with describing 1 as follows:

- The first number in the sequence describes 1 as 'One 1' which is then written as 11.
- The second number in the sequence describes the first number as 'Two 1s' which is then written as 21.
- Following this pattern, the third number is written as 'One 2, One 1', which is written as 1211.

Given a positive whole number n as input, output the nth number in the sequence.

Sample input: **Sample output:**

3   1211

Q3) A positive whole numbers is a palindrome if it reads the same from left to right as from right to left (e.g. 47174). Given any starting positive whole number with two or more digits which <u>isn't</u> a palindrome, we can instigate the following sequence to find a palindrome*:

Reverse its digits and add the resulting number to the original number. If the result is a palindrome then stop, otherwise repeat the process.

For example, start with 87. Applying this process, we obtain:

$87 + 78 = 165$

$165 + 561 = 726$

$726 + 627 = 1353$

$1353 + 3531 = \underline{4884}$

We can say the palindrome length cycle of 87 is 4 because it takes four additions in this process to produce a palindrome. Write a program to input a positive whole number less than 10,000 and output its palindrome length cycle.

Sample input:	**Sample output:**
87	4

Q1) In the card game Simple Pontoon (a simplified version of Pontoon), the playing cards are worth their face value for cards two to ten with Jack/Queen/King also each worth ten and an Ace worth eleven. The suit of any card is irrelevant to this problem and will be omitted. Two cards are dealt initially after which the player can choose to stick (finish the round) or twist (receive another card). If the value of the cards in the hand ever <u>exceed</u> 21 then the player is bust which means the hand is worth nothing. The possible outcomes for the each round are as follows:

<u>Round outcome</u> (best to worst)	<u>Meaning</u>
A	The best outcome - a hand of value 21 has been achieved from exactly two cards (e.g. a Jack and an Ace)
B	A five-card hand was achieved without going bust (e.g. 6, 3, 2, 5, 4)
Cxx	A hand value of 21 or less was achieved with four or less cards where xx represents the two digit value of the hand (e.g. C17 for a hand of 6,7,4 or C21 for a hand of 5,6,T)
D	A bust hand with cards worth over 21 (e.g. 7,9,9)

In this problem your input will represent a current deck to draw cards from. You must simulate a given number of rounds of the game and output the result from the <u>final</u> round, assuming that on each occasion the player adheres to the following rules:

- Always stick if the hand value is 17 or more
- Always stick if 5 cards are now in the hand and not bust
- Otherwise twist if not bust

Your program will receive two lines of input:

- A string of characters without spaces consisting of 2-9 and T (for 10), A (for Ace), J (for Jack), Q (for Queen) and K (for King) representing the current deck with the first character being the top of the deck. All cards in the deck will not necessarily be used.
- A whole number between 1 and 10 inclusive which indicates the number of rounds of the game to play.

You must output the outcome (using the table codes above) of the <u>final</u> round played.

Sample input: **Sample output:**

JA479K6T52   D

3

Sample explanation: In this example simulation then we are asked to play 3 rounds from the deck given which starts with a Jack at the top. In the first round the player receives a Jack and an Ace from the top of the deck and so outcome A has been achieved and the second round starts. In the second round the player receives a 4 and a 7 and so chooses to twist using the rules above and receives a 9. Their hand total is now 20 and so they stick and the outcome C20 is achieved. In the third and final round, the player receives a King and a 6 initially. Their hand is only worth 16 and so they twist and receive a ten and so are now bust and the outcome of the round is D and so D is printed out.

Q2) A game of leapfrog is played along a set of stepping stones by two teams, red and blue.

There are s stepping stones and p players on each team (where s > 2*p). The red team starts its players on the leftmost p stones and moves from left to right. The blue team starts its players on the rightmost p stones and moves from right to left. In this example below, there are three players for each team and seven stepping stones in total.

r r r _ b b b

A counting number c is given, and a moving number m is used, which is always given the value p to begin with. The game now commences following these rules:

- Counting from the left, player m moves where the player closest to the left side of the board is player 1 irrespective of colour.
- The player can do one of three things:
    - Move into an empty space in front of them (red moves right, blue moves left)
    - Jump a player in front of them (irrespective of colour) if there is a space on the other side of that player
    - Do nothing if neither of these conditions are met
    - A player may not jump off the stepping stones
- Now counting from the right, player m moves where the player closest to the right side of the board is player 1 irrespective of colour. The same moving rules apply.
- The value of m is then increased by c.
- If m is greater than p, m is reduced by p.

This is one round completed.

The input will consist of four lines each specifying a positive whole number as follows

- The number of stepping stones s
- The number of players on each team p
- The counting number c
- The number of <u>rounds</u> to play n

The program must output the state of the board after 1 round and after n <u>rounds</u>, each on a separate line. The board will be represented by a lowercase r for a red player, a lowercase b for a blue player and an underscore for an empty stepping stone.

Sample input: **Sample output:**
7   rrbr_bb
3
1   rrbrbb_

3

Sample explanation: The initial state of the board is rrr_bbb. m is set to 3 initially and so player 3 from the left (a red) gets to move one stone forward into the gap namely: rr_rbbb. Now counting three from the right, the blue player can jump this red and round 1 ends at rrbr_bb which is printed out. m is then increased 1 to be 4 but this is now over 3 and so it is reduced to 1 and round 2 begins. Counting 1 from the left (which is a red), the player can't move but the right most player can jump and so round 2 ends at rrbrbb_ and m is now increased to 2. For the third round neither the second player from the left or the second player from the right can move and so the board is unchanged and this is also then printed out because it is the end of the 3 rounds requested.