

Question 1) Write a Python function 'cowsBulls()' that will play the "cows and bulls" game with the user. The game works like this:

Randomly generate three numbers between 0 and 9 and store them in 'rand1', 'rand2', and 'rand3'. Ask the user to guess the three numbers in order (three input statements). For every digit that the user guessed correctly *in the correct place*, they have a "cow". For every digit the user guessed correctly *in the wrong place* is a "bull". Every time the user makes a guess, tell them how many "cows" and how many "bulls" they have. Once the user guesses the correct number, the game is over. Give the user 10 trials before he/she loses.

Example: say 'rand1' is 9, 'rand2' is 4, and 'rand3' is 2.

Trial 1: Say the user inputs the following three numbers (in order): 2, 4, and 5

Your program should print:

1 cow(s) and 1 bull(s) (1 cow for '4', and one bull for '2')

By now, the user doesn't know which number is the cow (correct guess), which number is the bull (correct number but incorrect order), and which number that is simply incorrect; so the user may re-orders the inputs and try again.

Trial 2: Say the user inputs these three numbers in order: 2, 5, and 4

Your program should print:

0 cow(s) and 2 bull(s)

The user should now know that 2 is not in its correct place and the user should also know that '2' is a bull (after thinking about for few minutes).

The user will continue to re-order the numbers or provide new numbers until he/she guesses the three numbers (in order), or until he/she runs out of trials.

Question 2) Many companies pay time-and-a-half for any hours worked above 40 in a given week. Write a Python function 'calcWages()' that takes (from the user) the number of hours worked and the hourly rate, and calculates the total wages for the week.

Question 3) A Fibonacci sequence is a sequence of numbers where each successive number is the sum of the previous two. The classic Fibonacci sequence begins: 1, 1, 2, 3, 5, 8, 13, ... Write a Python function 'calcFib()' that computes the n<sup>th</sup> Fibonacci number, where the value n is provided by the user. For example, if the user inputs n = 6, then the result should be 8.

Question 4) Make a one-player Rock-Paper-Scissors game. Write your Python code for this question in a function and call it 'rockPaperScissors()'

First, generate a play by the computer (random number from 1 to 3, 1 means rock, 2 means paper, and 3 means scissors). Then ask the player for his/her input, and finally display to the user whether he/she won or lost.

Have the user play the game 10 consecutive times and finally print the number of times the user won, and the number of times the computer won.

Remember the rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock

Question 5) Write a Python function 'series()', that asks the user for 'n', then adds n terms of the series:

$$\frac{4}{3} + \frac{7}{9} + \frac{10}{27} + \frac{13}{81} + \dots$$

Due: Monday, September 25th at midnight