CamCom Interview Questions (Coding) **passing marks (60)**

Question 1 (**40**)

You are given a matrix of size N x M containing characters '-', 'm', and 'p', where '-' represents an empty space, 'm' represents the starting position of Mario, and 'p' represents the position of the Princess. You need to implement a function rescue_princess(matrix) that takes the matrix as an input and returns a list of moves required by Mario to rescue the Princess in the least number of steps. The moves should be represented by 'LEFT', 'RIGHT', 'UP', and 'DOWN'. Mario can only move one step at a time in one of the four directions.

For example, given the following input:

Matrix: ["---","-m-","p–"]

```
-   -   -
-   m   -
p   -   -
```

The output should be:

```
DOWN
LEFT
```

Below is given function:

```
matrix=["---","-m-","p--"]

def process(matrix):
    '''
    Write your code here to give output of m=[x, y] position of mario and p=[x, y]
position of princess
    ex: for above given question returned values should be m=[1,1] p=[2,0]
    '''
    return m,p

m,p=process(matrix)
while True:
    print("Down\n" if m[0]-p[0]<0 else "UP\n" if m[0]-p[0]>0 else "", end="")
    m[0] += 1 if m[0]-p[0]<0 else -1 if m[0]-p[0]>0 else 0
    print("Left\n" if m[1]-p[1]>0 else "RIGHT\n" if m[1]-p[1]<0 else "", end="")
    m[1] += -1 if m[1]-p[1]>0 else 1 if m[1]-p[1]<0 else 0
    if m==p:
        break
```

CamCom Interview Questions (Coding) **passing marks (60)**

Question 2 (**30**)

Write a Python function that takes two integer arguments n and m, and simulates rolling two fair dice with n and m sides respectively. The function should then compute the probability of obtaining a sum of n and return the result as a float. If n and m are not positive integers, the function should raise a ValueError with a descriptive error message. Additionally, the function should check that n and m are within a reasonable range (e.g., less than or equal to 100), and should raise a ValueError with a descriptive error message if they are out of range.

Test your function with input values of n=6 and m=-4 using the provided simulate() function, and verify that the function raises a ValueError with a descriptive error message.

Output should be

0.16666666666666666

```python
def simulate(n,m):
    '''Write your code Here'''
    return probability


simulate(6,4)
```

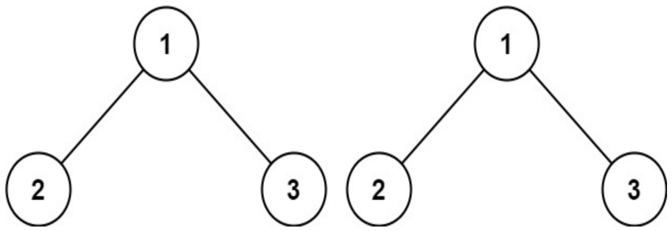
Question 3 (30)

You are given the roots of two binary trees p and q. Write a Python function is_same_tree(p, q) that checks if the two binary trees are structurally identical and have the same node values. If the two trees are identical, the function should return True. Otherwise, the function should return False.
The function should take in two arguments:

- p: The root node of the first binary tree, represented as a TreeNode object.
- q: The root node of the second binary tree, represented as a TreeNode object.

The TreeNode class is already provided for you.
Note:

- A binary tree is a tree data structure in which each node has at most two children, referred to as the left child and the right child.
- The TreeNode class has the following attributes:
  - val: an integer value that represents the node's value
  - left: a TreeNode object that represents the left child of the node
  - right: a TreeNode object that represents the right child of the node



Input: p = [1,2,3], q = [1,2,3]
Output: true

Test Case:
Input: p = [1,2], q = [1,null,2]

```
class Solution:
    def isSameTree(self, p: TreeNode, q: TreeNode) -> bool:
        '''Write Your code here'''
```

Question 4 (20)

Write a function two_sum(nums: List[int], target: int) -> List[int] that takes in a list of integers nums and an integer target, and returns the indices of the two numbers such that they add up to the target. Assume that there is exactly one solution and you may not use the same element twice.

```
# Test Case
assert two_sum([2, 7, 11, 15], 9) == [0, 1]
assert two_sum([3, 2, 4], 6) == [1, 2]
assert two_sum([3, 3], 6) == [0, 1]
```

Question 5 (40)

There is a manual QC process which happens. There is a portal from which each individual qc task is assigned. The portal needs to check how many qc persons are logged in and which of the logged in persons are free, as in not on a task, and automatically assign tasks. Once the task is finished the person will automatically get assigned the next task if any is pending. How would you architect this? I want to understand step by step the methodology you used to come to the final solution. Can you illustrate a basic API framework written in Python using Flask and MySql as the database.