Introduction to Programming
Coding Quiz

Question 1: Find the Missing Number

Description: You are given an array containing numbers from `1` to `n` with one missing number. Write a function that finds and returns the missing number.

Function Header:

```
int findMissing(int arr[], int size);
```

Explanation: The array should contain all numbers from 1 to n except one. Your task is to calculate the missing number using logic or a mathematical formula.

Test Cases:

- Input: {1, 2, 4, 5, 6} → Output: 3
- Input: {2, 3, 4, 5} → Output: 1
- Input: {1, 2, 3, 5, 6, 7, 8, 9, 10} → Output: 4

Starter Code:

```
// Write your code below

#include <iostream>
using namespace std;

int findMissing(int arr[], int size) {
    // Your code here
}

int main() {
    int arr1[] = {1, 2, 4, 5, 6};
    int arr2[] = {2, 3, 4, 5};
    int arr3[] = {1, 3, 2, 5, 6, 7, 8, 9, 10};

    cout << "Missing number (1): " << findMissing(arr1, 5) <<
       endl;
    cout << "Missing number (2): " << findMissing(arr2, 4) <<
       endl;
    cout << "Missing number (3): " << findMissing(arr3, 9) <<
       endl;
}
```

Question 2: Longest Common Ending

Description: Write a function that returns the **longest common ending (suffix)** between two strings.

Function Header:

```
string longestCommonEnding(string s1, string s2);
```

Explanation: The function compares the ends of both strings and finds the longest substring that appears at the end of both. If there is no common ending, return an empty string.

Test Cases:

- Input: ("multiplication", "ration") → Output: "ation"
- Input: ("potent", "tent") → Output: "tent"
- Input: ("skyscraper", "carnivore") → Output: ""

Starter Code:

```
#include <iostream>
#include <string>
using namespace std;

string longestCommonEnding(string s1, string s2) {
    // Your code here
}

int main() {
    cout << longestCommonEnding("multiplication", "ration") <<
       endl;
    cout << longestCommonEnding("potent", "tent") << endl;
    cout << longestCommonEnding("skyscraper", "carnivore") <<
       endl;
}
```

Question 3: Move All 'x' to the End

Description: Write a **recursive** function to move all occurrences of the character `'x'` to the end of a string.

Function Header:

```
string moveXToEnd(string s);
```

Explanation: Each recursive call should process one character. If the first character is `'x'`, move it to the end; otherwise, keep it at the front.

Hint: You can use the `substr()` function to get the substring of a given string.. The function `s.substr(1)` gives the substring of `s` starting from index 1. Eg if s= "computer" then s.substr(1) = "omputer".

Test Cases:

- Input: "axbxcxd" → Output: "abcdxxx"
- Input: "xxhixx" → Output: "hixxxx"
- Input: "xabc" → Output: "abcx"

Starter Code:

```
#include <iostream>
#include <string>
using namespace std;

string moveXToEnd(string s) {
    // Your code here
}

int main() {
    cout << moveXToEnd("axbxcxd") << endl;
    cout << moveXToEnd("xxhixx") << endl;
    cout << moveXToEnd("xabc") << endl;
}
```