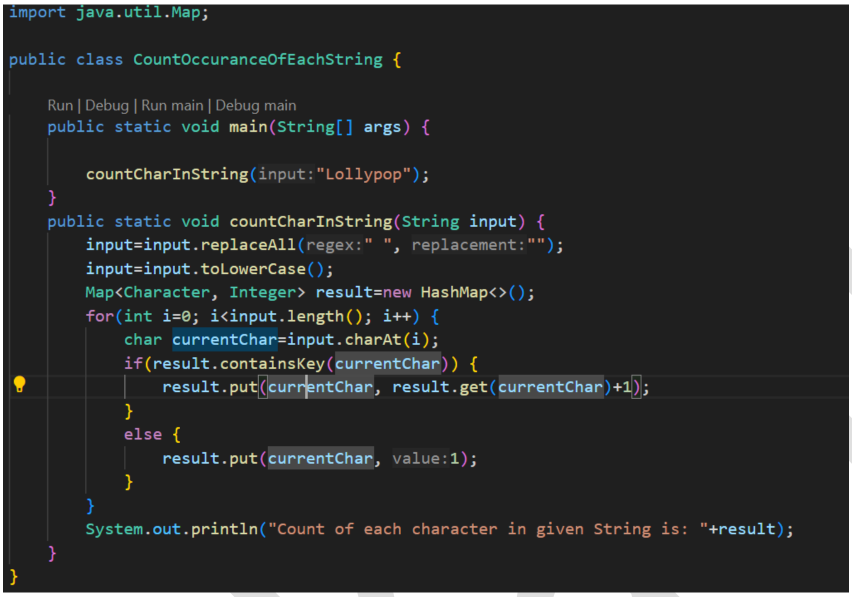
IMPORTANT JAVA INTERVIEW QUESTIONS

1. Character Occurrence Counter

□ Input: "Lollypop"

□ Output: L-3, O-2, P-2, y-1

☑ Write a program to count and print the frequency of each character.



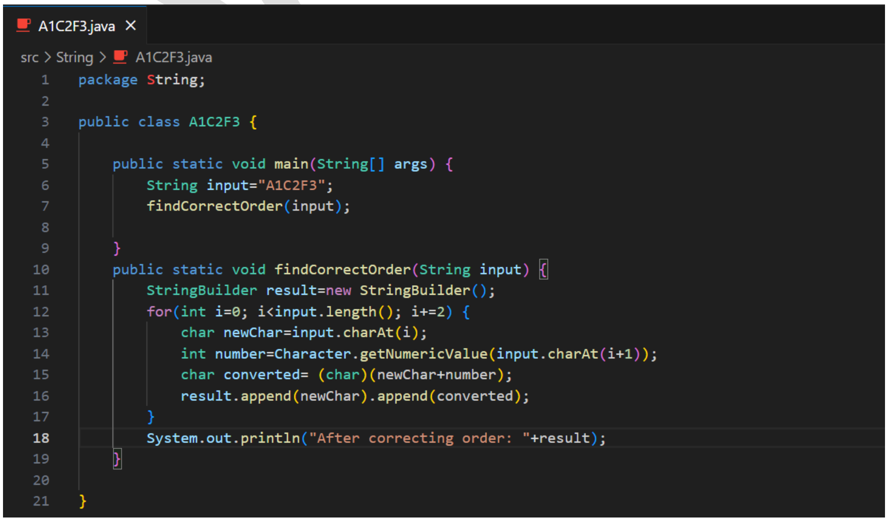
```java
import java.util.Map;

public class CountOccuranceOfEachString {

    public static void main(String[] args) {

        countCharInString(input:"Lollypop");
    }
    public static void countCharInString(String input) {
        input=input.replaceAll(regex:" ", replacement:"");
        input=input.toLowerCase();
        Map<Character, Integer> result=new HashMap<>();
        for(int i=0; i<input.length(); i++) {
            char currentChar=input.charAt(i);
            if(result.containsKey(currentChar)) {
                result.put(currentChar, result.get(currentChar)+1);
            }
            else {
                result.put(currentChar, value:1);
            }
        }
        System.out.println("Count of each character in given String is: "+result);
    }
}
```

2. ASCII Character Shift

□ Input: "A1C2F3"

□ Output: "ABCEFI"

💡 For every character followed by a digit, add the digit to the ASCII of the character and append the result.



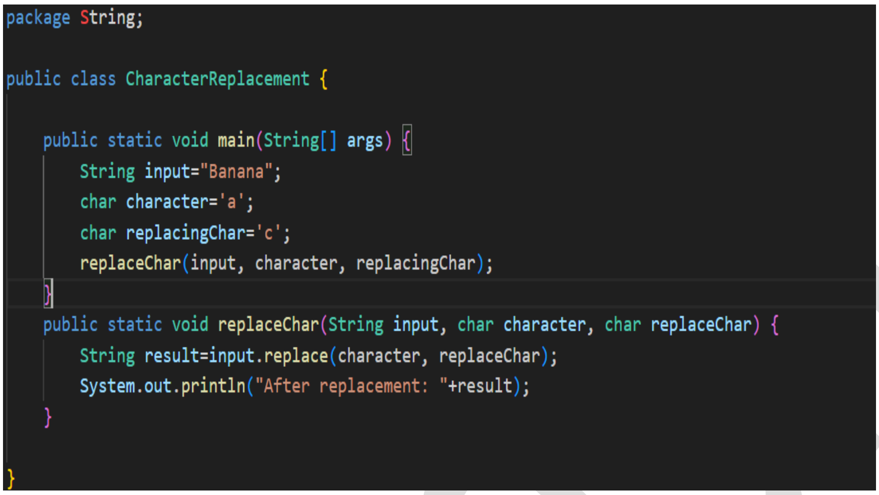
```java
package String;

public class A1C2F3 {

    public static void main(String[] args) {
        String input="A1C2F3";
        findCorrectOrder(input);

    }
    public static void findCorrectOrder(String input) {
        StringBuilder result=new StringBuilder();
        for(int i=0; i<input.length(); i+=2) {
            char newChar=input.charAt(i);
            int number=Character.getNumericValue(input.charAt(i+1));
            char converted= (char)(newChar+number);
            result.append(newChar).append(converted);
        }
        System.out.println("After correcting order: "+result);
    }

}
```

1. ## Character Replacer

□ Input: "Banana", Replace 'a' with 'c'

□ Output: "Bcncnc"

🔁 Replace all instances of a character with another.



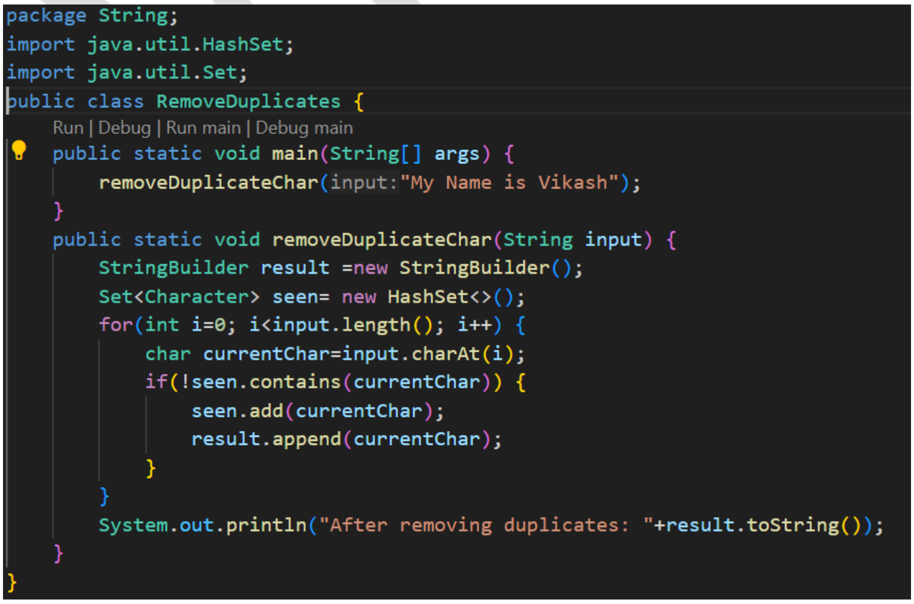
```java
package String;

public class CharacterReplacement {

    public static void main(String[] args) {
        String input="Banana";
        char character='a';
        char replacingChar='c';
        replaceChar(input, character, replacingChar);
    }
    public static void replaceChar(String input, char character, char replaceChar) {
        String result=input.replace(character, replaceChar);
        System.out.println("After replacement: "+result);
    }

}
```

2. ## Remove Duplicates (Keep Order)

□ Input: "My Name is Vikash"

□ Output: "My Naeiskvh"

□ Maintain the original order and remove repeating characters.



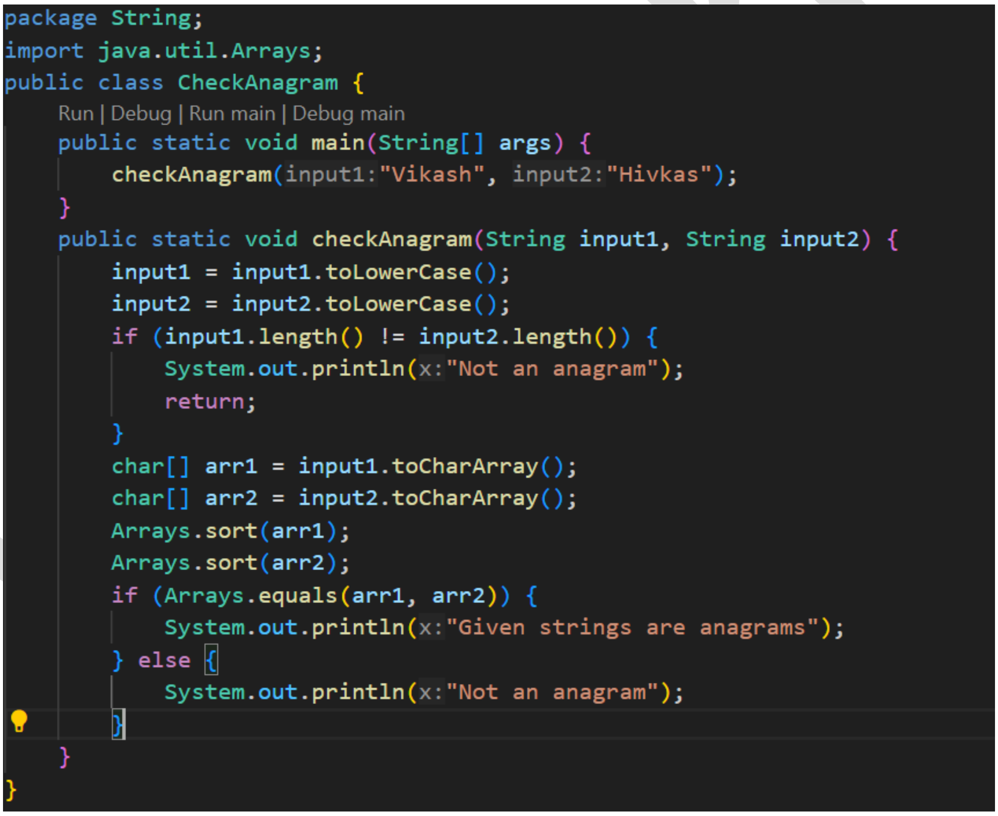
```java
package String;
import java.util.HashSet;
import java.util.Set;
public class RemoveDuplicates {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        removeDuplicateChar(input:"My Name is Vikash");
    }
    public static void removeDuplicateChar(String input) {
        StringBuilder result =new StringBuilder();
        Set<Character> seen= new HashSet<>();
        for(int i=0; i<input.length(); i++) {
            char currentChar=input.charAt(i);
            if(!seen.contains(currentChar)) {
                seen.add(currentChar);
                result.append(currentChar);
            }
        }
        System.out.println("After removing duplicates: "+result.toString());
    }
}
```

3. **Anagram Checker**

□ Input: "Vikash" vs "Hivkas"

□ Output: true

🔍 Check if two strings are anagrams — same letters, same frequency, different order.



```
package String;
import java.util.Arrays;
public class CheckAnagram {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        checkAnagram(input1:"Vikash", input2:"Hivkas");
    }
    public static void checkAnagram(String input1, String input2) {
        input1 = input1.toLowerCase();
        input2 = input2.toLowerCase();
        if (input1.length() != input2.length()) {
            System.out.println(x:"Not an anagram");
            return;
        }
        char[] arr1 = input1.toCharArray();
        char[] arr2 = input2.toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        if (Arrays.equals(arr1, arr2)) {
            System.out.println(x:"Given strings are anagrams");
        } else {
            System.out.println(x:"Not an anagram");
        }
    }
}
```