Lab-12-Logic Building

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a case_option parameter, as follows:

If case_option = 0, normal reversal of words i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "orpiW seigoloNhceT eroLagnaB".

If case_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw SeigOlonhcet EroIaGnab".

Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T, N, B and L.

Similarly, positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, E and G.

NOTE:

1. Only space character should be treated as the word separator i.e., "Hello World" should be treated as two separate words, "Hello" and "World". However, "Hello,World", "Hello;World", "Hello-World" or "Hello/World" should be considered as a single word.
2. Non-alphabetic characters in the String should not be subjected to case changes. For example, if case option = 1 and the original sentence is "Wipro TechNologies, Bangalore" the new reversed sentence should be "Orpiw ,seiGolonhceT Erolagnab". Note that comma has been treated as part of the word "Technologies," and when comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However, the words "Wipro and Bangalore" have changed to "Orpiw" and "Erolagnab".
3. Kindly ensure that no extra (additional) space characters are embedded within the resultant reversed String.

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolonhceT erolagnaB
2	Wipro Technologies, Bangalore	0	orpiW ,seigolonhceT erolagnaB
3	Wipro Technologies Bangalore	1	Orpiw Seigolonhcet Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolonhceT Erolagnab

---

CODE:

```
import java.util.Scanner;


public class WordReversal {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        String sentence = sc.nextLine();
        int caseOption = sc.nextInt();


        // Generate and display the modified sentence
        String result = reverseWords(sentence, caseOption);
        System.out.println(result);


        sc.close();
    }


    public static String reverseWords(String sentence, int case_option) {
        String[] words = sentence.split(" ");
```

```
    StringBuilder modifiedSentence = new StringBuilder();

    for (int i = 0; i < words.length; i++) {
        String word = words[i];
        StringBuilder reversedWord = new StringBuilder();

        for (int j = word.length() - 1; j >= 0; j--) {
            reversedWord.append(word.charAt(j));
        }

        if (case_option == 1) {
            for (int j = 0; j < word.length(); j++) {
                char originalChar = word.charAt(j);
                char reversedChar = reversedWord.charAt(j);

                if (Character.isUpperCase(originalChar)) {
                    reversedWord.setCharAt(j, Character.toUpperCase(reversedChar));
                } else if (Character.isLowerCase(originalChar)) {
                    reversedWord.setCharAt(j, Character.toLowerCase(reversedChar));
                }
            }
        }

        modifiedSentence.append(reversedWord);
        if (i < words.length - 1) {
            modifiedSentence.append(" ");
        }
    }

    return modifiedSentence.toString();
}
```

}

OUTPUT:

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB	orpiW ,seigolonhceT erolagnaB	✓
✓	Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab	Orpiw Seigolonhcet Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

Note:

1. **Array size ranges from 1 to 10.**
2. **All the array elements are lower case alphabets.**
3. **Atleast one common alphabet will be found in the arrays.**

Example 1:

input1: {'a', 'b', 'c'}

input2: {'b', 'c'}

output: 8

Explanation:

'b' and 'c' are present in both the arrays.

ASCII value of 'b' is 98 and 'c' is 99.

98 + 99 = 197

1 + 9 + 7 = 17

1 + 7 = 8

CODE:

```
import java.util.HashSet;
import java.util.Scanner;


public class CommonCharAsciiSum {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);


      char[] input1 = sc.nextLine().replace(" ", "").toCharArray();


      char[] input2 = sc.nextLine().replace(" ", "").toCharArray();
```

```
    int result = getSingleDigitAsciiSum(input1, input2);
    System.out.println(result);

    sc.close();
}

public static int getSingleDigitAsciiSum(char[] input1, char[] input2) {
    HashSet<Character> set1 = new HashSet<>();
    HashSet<Character> commonChars = new HashSet<>();

    // Add characters from input1 to the set1
    for (char c : input1) {
        set1.add(c);
    }

    // Find common characters by checking presence in input2
    for (char c : input2) {
        if (set1.contains(c)) {
            commonChars.add(c);
        }
    }

    // Calculate the ASCII sum of common characters
    int sum1 = 0;
    for (char c : commonChars) {
        sum1 += (int) c;
    }

    // Reduce sum1 to a single-digit sumreturn reduceToSingleDigit(sum1);
}
```

```
    private static int reduceToSingleDigit(int num) {
        while (num >= 10) {
            int tempSum = 0;
            while (num > 0) {
                tempSum += num % 10;
                num /= 10;
            }
            num = tempSum;
        }
        return num;
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (00000000000000000000000000).

The sequence of 0's in the encoded form are separated by a single 1 which helps to distinguish between 2 letters.

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 0000100000000000000000010000000000010000000001000000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case.

CODE:

```
import java.util.Scanner;
public class Decoder {
public static String decode(String input) {
    String[] zeroSequences = input.split("1");
    StringBuilder decodedWord = new StringBuilder();
    for (String zeros : zeroSequences) {
       int zeroCount = zeros.length();
       if (zeroCount > 0 && zeroCount <= 26) {
          char letter = (char) ('Z' - zeroCount + 1);
```

```
                decodedWord.append(letter);
            }
        }
        return decodedWord.toString();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.nextLine();
        System.out.println(decode(input));
        scanner.close();
    }
}
```

OUTPUT:

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000001000000000001000000000100000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓