

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-12-Introduction to I/O, I/O Operations, Object Serialization](#) / [Lab-12-Logic Building](#)

| | |
|------------------|------------------------------------|
| Status | Finished |
| Started | Monday, 18 November 2024, 10:07 PM |
| Completed | Monday, 18 November 2024, 11:01 PM |
| Duration | 54 mins 40 secs |

Question 1

Correct

Marked out of 5.00

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 SeigOlönhcet ErolaGnab".

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:

- 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.
- 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 ,seigOlönhcet 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".
- 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 SeigOlönhcet Erolagnab |
| 4 | Wipro Technologies, Bangalore | 1 | Orpiw ,seigolonhceT Erolagnab |

For example:

| Input | Result |
|------------------------------------|-------------------------------|
| Wipro Technologies Bangalore 0 | orpiW seigolonhceT erolagnaB |
| Wipro Technologies, Bangalore 0 | orpiW ,seigolonhceT erolagnaB |
| Wipro Technologies Bangalore 1 | Orpiw SeigOlönhcet Erolagnab |
| Wipro Technologies, Bangalore 1 | Orpiw ,seigolonhceT Erolagnab |

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class SentenceReverser {
4     public static String reverseWords(String sentence, int caseOption) {
5         String[] words = sentence.split(" ");
6         StringBuilder reversedSentence = new StringBuilder();
7
8         for (String word : words) {
9             StringBuilder reversedWord = new StringBuilder(word).reverse();
10
11             if (caseOption == 1) {
12                 for (int i = 0; i < word.length(); i++) {
13                     char originalChar = word.charAt(i);
14                     char reversedChar = reversedWord.charAt(i);
15
16                     if (Character.isUpperCase(originalChar)) {
17                         reversedWord.setCharAt(i, Character.toUpperCase(reversedChar));
18                     } else if (Character.isLowerCase(originalChar)) {

```

| | 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! ✓

Question 2

Correct

Marked out of 5.00

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.

$98 + 99 = 197$

$1 + 9 + 7 = 17$

$1 + 7 = 8$

For example:

| Input | Result |
|-------|--------|
| a b c | 8 |
| b c | |

Answer: (penalty regime: 0 %)

```

1 import java.util.HashSet;
2
3 public class CommonAlphabetSum {
4
5     public static int singleDigitSum(char[] input1, char[] input2) {
6         // Step 1: Find common characters
7         HashSet<Character> set1 = new HashSet<>();
8         HashSet<Character> commonChars = new HashSet<>();
9
10        for (char c : input1) {
11            set1.add(c); // Add all characters of input1 to the set
12        }
13
14        for (char c : input2) {
15            if (set1.contains(c)) {
16                commonChars.add(c); // Add to common set if present in input1
17            }
18        }
19
20        // Step 2: Calculate sum of ASCII values of common characters
21        int sum1 = 0;
22        for (char c : commonChars) {
23            sum1 += (int) c; // Add ASCII value of each common character
24        }
25
26        // Step 3: Calculate single-digit sum of sum1
27        return calculateSingleDigitSum(sum1);
28    }
29
30    private static int calculateSingleDigitSum(int num) {
31        while (num > 9) { // Continue until single digit

```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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 (00000000000000000000000000000000).

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: 0000100000000000000000001000000000001000000000100000000000001

The decoded string (original word) will be: WIPRO

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

For example:

| Input | Result |
|---------------------------------------------------------------|--------|
| 010010001 | ZYX |
| 0000100000000000000000001000000000001000000000100000000000001 | WIPRO |

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 public class DecodeSequence {
3     public static String decodeString(String encoded) {
4         // Split the encoded string by '1' to separate each sequence of 0's
5         String[] parts = encoded.split("1");
6         StringBuilder decodedWord = new StringBuilder();
7
8         // Process each sequence of 0's
9         for (String part : parts) {
10             int countZeros = part.length(); // Count the number of 0's
11             char decodedChar = (char) ('Z' - (countZeros - 1)); // Decode to character
12             decodedWord.append(decodedChar); // Append to the result
13         }
14
15         return decodedWord.toString();
16     }
17
18     public static void main(String[] args) {
19
20         Scanner scan = new Scanner(System.in);
21         String str = scan.nextLine();
22         System.out.println(decodeString(str));
23     }
24 }
25 }
```

Passed all tests! ✓

Jump to...

Identify possible words ►