

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

Status	Finished
Started	Friday, 22 November 2024, 8:11 PM
Completed	Friday, 22 November 2024, 8:54 PM
Duration	43 mins 29 secs

Question 1

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 (000000000000000000000000000000).

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.Scanner;
2
3 public class DecodeString {
4
5     // Method to decode the input string
6     public static String decode(String encoded) {
7         // Split the string by '1' to get each group of zeros
8         String[] parts = encoded.split("1");
9         StringBuilder decodedString = new StringBuilder();
10
11         // Loop through each group of zeros
12         for (String part : parts) {
13             int numOfZeros = part.length();
14
15             if (numOfZeros > 0) {
16                 // Calculate the corresponding letter
17                 char letter = (char) ('Z' - numOfZeros + 1);
18                 decodedString.append(letter);
19             }
20         }
21     }
22 }

```

```

20     }
21
22     return decodedString.toString();
23 }
24
25 // Main method to handle user input and output
26 public static void main(String[] args) {
27     Scanner scanner = new Scanner(System.in);
28
29     // Take input from the user
30     String encodedInput = scanner.nextLine();
31
32     // Decode and output the result
33     String decodedOutput = decode(encodedInput);
34     System.out.println(decodedOutput);
35
36     scanner.close();
37 }
38 }

```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000001000000000001000000000010000000000001	WIPRO	WIPRO	✓

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 b c	8

Answer: (penalty regime: 0 %)

```

1 import java.util.HashSet;
2
3 public class CommonAlphabets {
4
5     // Method to calculate the single-digit sum of a number
6     public static int calculateSingleDigitSum(int sum) {
7         while (sum >= 10) {
8             sum = sumOfDigits(sum);
9         }
10        return sum;
11    }
12
13    // Method to calculate the sum of digits of a number
14    public static int sumOfDigits(int num) {
15        int sum = 0;
16        while (num > 0) {
17            sum += num % 10;
18            num /= 10;
19        }
20        return sum;
21    }
22 }
```

```

22
23 // Method to calculate the sum of ASCII values of common characters and reduce it to a single digit
24 public static int commonAlphabetSum(char[] input1, char[] input2) {
25     HashSet<Character> set1 = new HashSet<>();
26     HashSet<Character> set2 = new HashSet<>();
27
28     // Add characters from the first array to set1
29     for (char c : input1) {
30         set1.add(c);
31     }
32
33     // Add characters from the second array to set2
34     for (char c : input2) {
35         set2.add(c);
36     }
37
38     // Retain only common characters in set1
39     set1.retainAll(set2);
40
41     // Calculate the sum of ASCII values of the common characters
42     int sum = 0;
43     for (char c : set1) {
44         sum += (int) c;
45     }
46
47     // Reduce the sum to a single digit and return
48     return calculateSingleDigitSum(sum);
49 }
50
51 public static void main(String[] args) {
52     char[] input1 = {'a', 'b', 'c'};

```

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

Passed all tests! ✓

Question 3

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 SeigOlonhceT 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:

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

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB
Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB
Wipro Technologies Bangalore 1	Orpiw SeigolonhceT Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class WordReversal {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         // Read the sentence and case option
8         String sentence = sc.nextLine();
9         int caseOption = sc.nextInt();

```

```

10 // Exception handling
11 // Generate and display the modified sentence
12 String result = reverseWords(sentence, caseOption);
13 System.out.println(result);
14
15 sc.close();
16 }
17
18 public static String reverseWords(String sentence, int caseOption) {
19     // Split the sentence into words
20     String[] words = sentence.split(" ");
21     StringBuilder modifiedSentence = new StringBuilder();
22
23     for (int i = 0; i < words.length; i++) {
24         String word = words[i];
25         StringBuilder reversedWord = new StringBuilder();
26
27         // Reverse the characters in the word
28         for (int j = word.length() - 1; j >= 0; j--) {
29             reversedWord.append(word.charAt(j));
30         }
31
32         // Maintain the original case if caseOption == 1
33         if (caseOption == 1) {
34             for (int j = 0; j < word.length(); j++) {
35                 char originalChar = word.charAt(j);
36                 char reversedChar = reversedWord.charAt(j);
37
38                 if (Character.isUpperCase(originalChar)) {
39                     reversedWord.setCharAt(j, Character.toUpperCase(reversedChar));
40                 } else if (Character.isLowerCase(originalChar)) {
41                     reversedWord.setCharAt(j, Character.toLowerCase(reversedChar));
42                 }
43             }
44         }
45
46         // Append the reversed word to the result
47         modifiedSentence.append(reversedWord);
48
49         // Add a space between words (except after the last word)
50         if (i < words.length - 1) {
51             modifiedSentence.append(" ");
52         }
53     }
54 }

```

	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! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ▶

