

CS23333-Object Oriented Programming Using Java-2023

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

Quiz navigation



[Show one page at a time](#)
[Finish review](#)

Question **1**

Correct

Marked out of 5.00

Flag question

Status	Finished
Started	Saturday, 16 November 2024, 9:59 PM
Completed	Saturday, 16 November 2024, 10:33 PM
Duration	34 mins 16 secs

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: 0000100000000000000000001000000000010000000010000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
000010000000000000000000100000000010000000010000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static String decode(String str) {
5         StringBuilder decodedWord = new StringBuilder();
6         String[] groups = str.split("1");
7         for (String group : groups)
8         {
9             int zeroCount = group.length();
10            if (zeroCount > 0) {
11                char letter = (char) ('Z' - (zeroCount - 1));
12                decodedWord.append(letter);
13            }
14        }
15        return decodedWord.toString();
16    }
17
18    public static void main(String[] args) {
19        Scanner sc = new Scanner(System.in);
20        String input = sc.nextLine();
21
22        String result = decode(input);
23        System.out.println(result);
24
25        sc.close();
26    }
27 }
28 }
```

Input	Expected	Got
010010001	ZYX	ZYX
000010000000000000000000100000000010000000010000000000001	WIPRO	WIPRO

Passed all tests!

Question **2**

Correct

Marked out of 5.00

Flag question

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:

- One 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 ,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 public class Main {
3     public static String rev(String str1, int caseOption) {
4         String[] eachWords = str1.split(" ");
5         StringBuilder revString = new StringBuilder();
6         for (String word : eachWords) {
7             StringBuilder r = new StringBuilder(word).reverse();
8             if (r.charAt(r.length() - 1) == ',') {
9                 char lastChar = r.charAt(r.length() - 1);
10                r.deleteCharAt(r.length() - 1);
11                r.reverse();
12                r.append(lastChar);
13            }
14            if (caseOption == 1) {
15                for (int i = 0; i < word.length(); i++) {
16                    if (Character.isUpperCase(word.charAt(i))) {
17                        r.setCharAt(i, Character.toUpperCase(r.charAt(i)));
18                    } else if (Character.isLowerCase(word.charAt(i))) {
19                        r.setCharAt(i, Character.toLowerCase(r.charAt(i)));
20                    }
21                }
22            }
23            revString.append(r).append(" ");
24        }
25        return revString.toString().trim();
26    }
27    public static void main(String[] args) {
28        Scanner sc = new Scanner(System.in);
29        String strGiven = sc.nextLine();
30        int caseOption = sc.nextInt();
31        String result = rev(strGiven, caseOption);
32        System.out.println(result);
33        sc.close();
34    }
35 }
```

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 **3**
Correct
Marked out of
5.00
[Flag question](#)

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 public class CommonCharSum {
```

```

2  public static int digitalRoot(int sum) {
3
4      while (sum >= 10) {
5          int tempSum = 0;
6          while (sum > 0) {
7              tempSum += sum % 10;
8              sum /= 10;
9          }
10         sum = tempSum;
11     }
12     return sum;
13 }
14
15 public static void main(String[] args) {
16
17     char[] input1 = {'a', 'b', 'c'};
18     char[] input2 = {'b', 'c'};
19
20     int sum1 = 0;
21
22     for (int i = 0; i < input1.length; i++) {
23         for (int j = 0; j < input2.length; j++) {
24             if (input1[i] == input2[j]) {
25                 sum1 += (int) input1[i];
26                 break;
27             }
28         }
29     }
30     int result = digitalRoot(sum1);
31     System.out.println(result);
32 }
33 }

```

	Input	Expected	Got
	a b c	8	8
	b c		

Passed all tests!

[Finish review](#)

[Lab-12-MCQ](#)

Jump to...

[Identify possible words](#)