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

Status	Finished
Started	Saturday, 9 November 2024, 9:46 PM
Completed	Saturday, 9 November 2024, 10:24 PM
Duration	37 mins 47 secs

Question **1**

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 %)

```
public class CommonCharsSum {

    public static void main(String[] args) {
        char[] input1 = {'a', 'b', 'c'};
        char[] input2 = {'b', 'c'};

        System.out.println(getSingleDigitSum(input1, input2));
    }

    public static int getSingleDigitSum(char[] input1, char[] input2) {
        // Step 1: Find common characters
        int sum = 0;
        for (char c1 : input1) {
            for (char c2 : input2) {
                if (c1 == c2) {
                    // Step 2: Add ASCII value of common characters
                    sum += (int) c1;
                }
            }
        }

        // Step 3: Reduce the sum to a single digit
        while (sum >= 10) {
            sum = sumDigits(sum);
        }

        return sum;
    }

    // Helper method to calculate sum of digits of a number
    public static int sumDigits(int num) {
        int sum = 0;
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }
        return sum;
    }
}

```

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

Passed all tests! ✓

Question **2**

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 %)

```
import java.util.Scanner;

public class SentenceTransformer {

    public static void main(String[] args) {
        // Create a scanner to read user input
        Scanner scanner = new Scanner(System.in);

        // Ask the user for a sentence
        // Ask the user for a sentence

        String sentence = scanner.nextLine();

        // Ask the user for the case option
        int caseOption = scanner.nextInt();

        // Transform the sentence based on user input
        String transformedSentence = transformSentence(sentence, caseOption);
        System.out.println(transformedSentence);

        // Close the scanner
        scanner.close();
    }

    public static String transformSentence(String sentence, int caseOption) {
        // Split the sentence into individual words
        String[] words = sentence.split(" ");
        StringBuilder result = new StringBuilder();

        for (String word : words) {
            // Reverse the word
            String reversedWord = new StringBuilder(word).reverse().toString();

            if (caseOption == 1) {
                // Preserve the case positions based on the original word
                reversedWord = preserveCaseWithPosition(word, reversedWord);
            }

            // Append the reversed word to the result with a space
            result.append(reversedWord).append(" ");
        }

        // Remove the last trailing space and return the final result
        return result.toString().trim();
    }

    // Function to preserve the original case position while reversing
    private static String preserveCaseWithPosition(String original, String reversed) {
        StringBuilder modifiedWord = new StringBuilder();

        // Loop through each character of the reversed word and the original word
        for (int i = 0; i < original.length(); i++) {
            char originalChar = original.charAt(i);
```

	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

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: 00001000000000000000000100000000000100000000010000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
`010010001`	`ZYX`
`00001000000000000000000100000000000100000000010000000000001`	`WIPRO`

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;

public class BinaryDecoder {

    public static void main(String[] args) {
        // Create a Scanner object for reading input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the encoded string
        String encodedString = scanner.nextLine();

        // Decode the string and print the result
        String decodedWord = decodeBinaryString(encodedString);
        System.out.println( decodedWord);

        // Close the scanner
        scanner.close();
    }
    public static String decodeBinaryString(String encodedString) {
        // Split the encoded string by '1' to get sequences of '0's
        String[] letterCodes = encodedString.split("1");
        StringBuilder decodedWord = new StringBuilder();

        for (String letterCode : letterCodes) {
            int zeroCount = letterCode.length(); // Count the number of '0's in the sequence
            if (zeroCount > 0) {
                // Calculate the corresponding letter from 'Z' to 'A'
                char decodedChar = (char) ('Z' - zeroCount + 1);
                decodedWord.append(decodedChar);
            }
        }

        return decodedWord.toString();
    }
}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	0000100000000000000000000010000000000001000000000100000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

◄ Lab-12-MCQ

Jump to...

Identify possible words ►