

# Advanced Constructs: Advanced Function Concepts-2 Assignment Solutions



## Assignment Solutions

**1. Bikers A and B started the journey and A having X litres of petrol and B having Y litres of petrol. K and L are the km per litre for A and B bikes.**

Let us consider halt as an array a1, a2, a3, .... an where n is the length of the array. a represent distance from one halt to another.

Write a program to find each biker will cross how many halts based on the petrol quantity and the litre per km.

**Sample Input:**

```
10 12
20 15
40 98 134 179 211 248
```

**Sample Output:**

```
2 2
```

**Solution:**

```
const Bikers = (x, y, k, l, arr) => {
    const aKm = x * k;
    const bKm = y * l;
    let aCount = 0, bCount = 0,
        aTotal = 0, bTotal = 0;
    arr.map(item => {
        aTotal += parseInt(item);
        bTotal += parseInt(item);
        if (aKm > aTotal) {
            aCount++;
        }
        if (bKm > bTotal) {
            bCount++
        }
    });
    return `${aCount} ${bCount}`;
}
const inp = `10 12
20 15
40 98 134 179 211 248`;

const [ x, y ] = inp.split('\n')[0].split(' ');
const [ k, l ] = inp.split('\n')[1].split(' ');
const arr = inp.split('\n')[2].split(' ');

console.log(Bikers(x, y, k, l, arr));
```

To run the code live and test click [here](#).

**2. Write a program to get an input ('How are you') and by default it should be good in the prompt text box.**

**Solution:**

```
let greet = prompt("How are you?", "I am Good");  
console.log(greet);
```

**Output:**

"I am good"

To run the code live and test click [here](#).

**3. Assume that the input will be a word with odd letter count. Print the output as given below. start with middle letter from first line. Next line two letter from middle . Continue still you print all letters in last line. Then start with the first letter and continue for the remaining letters.**

**Constraint**

Input string length should be odd number not even

**Input Format**

The input should be string separated by space

**Output Format**

Start with middle letter from first line. Next line two letter from middle . Continue still you print all letters in last line. Then start with the first letter and continue for the remaining letters.

**Sample Input 1:**

W E L C O M E

**Explanation:** Start with middle letter 'C' from first line. Next line two letter 'C O' from middle . Continue still you print all letters in last line. Then start with the first letter and continue for the remaining letters.

**Output :**

```
 C  
 CO  
 COM  
 COME  
 COMEW  
 COMEW  
 COMEWEL
```

## Solution:

```
const oddLetter = letter => {

    const letterArr = letter.split(' ');
    const letterLen = letterArr.length;
    const middle = (letterLen -1) / 2;
    const newLetter = letterArr.slice(middle).concat(letterArr.slice(0,
middle))
    const arr = [];
    letterArr.forEach(element => {
        arr.push(newLetter.join(' '));
        newLetter.pop();
        newLetter.unshift(' ');
    });
    arr.reverse();
    return arr.join('\n');
}

console.log(oddLetter('R E L E V E L'))
```

To run the code live and test click [here](#).

**4. Seetha is working as a professor in a college and she is having 100+ student records and she wants to convert student's names into uppercase. Write a program to help Seetha to Change the student name into uppercase.**

## Constraint

Input should be string not an number or any other data type  
Input string should be lowercase or uppercase letters

## Input Format

The input should be string separated by new line

## Output Format

Sort the given set of strings and print each string one by one.

## Sample Input 1:

divya  
seetha  
shanmugi  
mohan khan  
mohan tata

## Output:

DIVYA  
SEETHA

SHANMUGI  
MOHAN KHAN  
MOHAN TATA

### Solution:

```
const LOWER_ALPHABET = 'abcdefghijklmnopqrstuvwxyz';
const UPPER_ALPHABET = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

const upperCase = (arr) => {
    const lowerArr = LOWER_ALPHABET.split('');
    const upperArr = UPPER_ALPHABET.split('');
    const lowerObj = {};
    const upperObj = {};
    const outputArr = [];
    lowerArr.map((item, index) => lowerObj[item] = (index + 1));
    upperArr.map((item, index) => upperObj[index + 1] = item);

    arr.map(inp => {
        let newStr = '';
        inp.split('').map(item => {
            if (lowerObj[item]) {
                newStr += upperObj[lowerObj[item]];
            }
            else {
                newStr += item;
            }
        })
        outputArr.push(newStr)
    })

    return outputArr;
}

console.log(upperCase(["divya",
"seetha",
"shanmugi",
"mohan khan",
"mohan tata"])
))
```

To run the code live and test click [here](#).