

Problem solving practice

These challenges will help you practice problem solving as well as using programming fundamentals such as variables, data types, control flow, functions, arrays and objects.

Prompts

1. Highest and Lowest Given a string of space-separated digits representing integers, return the highest and lowest integers.

```
highestAndLowest("7 12 29 4"); // [4, 29]
highestAndLowest("-700 102 90 -4"); // [ -700, 102 ]
highestAndLowest("-2 -3 -1"); // [ -3, -1 ]
highestAndLowest("07 019 12 01"); // [ 1, 19 ]
```

2. Is it an isogram? Determine if a string is an isogram (case-insensitive). An isogram has no repeating letters. The string will be made up only of English letters.

```
isIsogram("abc"); // true
isIsogram("ABbC"); // false
isIsogram(""); // true
isIsogram("Thequickbrownfoxjumpedoverthelazyreddog"); // false
```

3. Replace letters Given a string, return a string with all a's replaced with 4s, all e's replaced with 3s and all l's replaced with 1s.

```
replaceLetters("Hello World"); //H311o World
const result10 = replaceLetters("Earth"); //34rth
const result11 = replaceLetters(""); //""
const result12 = replaceLetters("All"); //411
```

4. Inventory Merger You have two objects representing warehouse inventories.

- `currentInv` maps item names → current stock (non-negative integers).
- `newShip` maps item names → newly arrived stock (non-negative integers).

Write a function `mergeInventories(currentInv, newShip)` that returns a brand-new object where each key is an item name and its value is the sum of stocks from both inputs.

If an item appears in only one of the inputs, include it with whatever count it has.

```
mergeInventories({ apples: 5, oranges: 10 }, { bananas: 3, apples: 2 });
// → { apples: 7, oranges: 10, bananas: 3 }
```

```
mergeInventories({ pen: 10, notebook: 5 }, { notebook: 7, eraser: 2 });  
// → { pen: 10, notebook: 12, eraser: 2 }  
  
mergeInventories({}, { screws: 100 });  
// → { screws: 100 }
```

5. Square Every Digit Given an integer, square every digit and concatenate them.

```
squareDigit(44); //1616  
squareDigit(11); //11  
squareDigit(22); //44  
squareDigit(317); //9149  
squareDigit(307); //9049
```