

Bài tập bài 07: Javascript cơ bản (Tiết 3)

Câu 01. Kiểm tra số nguyên dương.

- Đề bài:
 - Theo lý thuyết thì số nguyên dương là tập hợp những số chia hết cho 1 và lớn hơn 0. Tức là dãy số 1, 2, 3, 4, 5, 6 ... chính là các số nguyên dương.
 - Dùng hàm prompt để tạo ô nhập và lấy ra giá trị từ ô nhập đó.
 - Viết một hàm kiểm tra số nhập vào có phải số nguyên dương hay không?
 - Nếu đúng alert ra YES.
 - Nếu sai alert ra NO.
- Đáp án:

```
function laSoNguyenDuong(value) {  
    return value > 0 && value % 1 == 0;  
};  
  
var input = prompt("Nhập một số bất kỳ", "");  
input = parseFloat(input);  
  
if (isNaN(input)) {  
    alert("Vui lòng nhập một số!");  
} else {  
    var result = laSoNguyenDuong(input);  
    if (result == true) {  
        alert("YES");  
    } else {  
        alert("NO");  
    }  
}
```

Câu 02: Kiểm tra có phải là một hình?

- Đề bài:
 - Một hình gồm n cạnh và n góc. Nhập vào số n và một mảng gồm n phần tử chứa các góc (độ) của một hình (hình tam giác, hình vuông,...).
 - Ví dụ:
 - Hình tam giác có 3 cạnh, và các góc lần lượt là 80°, 70° và 30°.
 - Vậy ta có n = 3 và array=[80, 70, 30]
- Cho người dùng nhập vào n và một mảng gồm n phần tử là các góc. Viết một hàm kiểm tra xem từ số n và mảng đó có thỏa mãn là một hình vẽ hay không?
 - Nếu thỏa mãn trả ra **true**
 - Nếu không thỏa mãn trả ra **false**
- Lưu ý:
 - Trả về **false** nếu **n < 3**.
 - Tổng các góc trong là **(n - 2) x 180°**.

- Ví dụ:
 - $n = 4$ và $\text{array} = [90, 90, 90, 90] \rightarrow \text{true}$
 - $n = 3$ và $\text{array} = [20, 20, 140] \rightarrow \text{true}$
 - $n = 1$ và $\text{array} = [21] \rightarrow \text{false}$ // vì n phải lớn hơn 2
 - $n = 5$ và $\text{array} = [500, 0, 20, 10, 10] \rightarrow \text{false}$ // vì tổng các góc không thể lớn hơn 180° và cũng không có góc nào nhỏ hơn hoặc bằng 0° .

- **Đáp án:**

```

● const sum = (array) => {
●   const total = array.reduce((total, number) => total + number, 0);
●   return total;
● }
●
● const isShapePossible = (n, angles) => {
●   if (n < 3) return false;
●
●   if (angles.some((angle) => angle <= 0 || angle >= 180)) return false;
●
●   return sum(angles) === (n - 2) * 180;
● };
●
● const test1 = isShapePossible(4, [90, 90, 90, 90]);
● console.log(test1); // true
●
● const test2 = isShapePossible(3, [20, 20, 140]);
● console.log(test2); // true
●
● const test3 = isShapePossible(1, [21]);
● console.log(test3); // false
●
● const test4 = isShapePossible(5, [500, 0, 20, 10, 10]);
● console.log(test4); // false

```

Câu 03: Viết tắt câu

- **Đề bài:**
 - Tạo một hàm nhận vào một câu văn và trả về chữ viết tắt của câu văn đó. Lấy tắt cả các từ có độ dài lớn hơn hoặc bằng n ký tự và trả về chữ cái đầu tiên của mỗi từ, được viết hoa.
 - Ví dụ:
 - "Xin chào! Tôi tên là Nam." \rightarrow "XCTTLN"
 - "Xin chào! Tôi tên là Nam." và $n = 3 \rightarrow$ "XCTTN", Chữ "là" bị bỏ qua, vì có 2 ký tự, nhỏ hơn $n = 3$
 - "Xin chào! Tôi tên là Nam." và $n = 4 \rightarrow$ "CN", Các chữ có 3 ký tự trở xuống bị bỏ, vì $n = 4$
- **Đáp án:**
 - Cách 1: Viết đầy đủ.
 - ```
const abbreviate = (string, n = 0) => {
 let array = string.split(" ");
 let arrayFilter = array.filter((word) => word.length >= n);
 let arrayChar = arrayFilter.map((item) => {
 return item[0];
 });
```

```

o let stringChar = arrayChar.join("");
o let stringUpper = stringChar.toUpperCase();
o return stringUpper;
o };
o
o const test1 = abbreviate("Xin chào! Tôi tên là Nam.");
o console.log(test1); // XCTTLN
o
o const test2 = abbreviate("Xin chào! Tôi tên là Nam.", 3);
o console.log(test2); // XCTTN --> Chữ "là" bị bỏ qua, vì có 2 ký
tự, nhỏ hơn n = 3
o
o const test3 = abbreviate("Xin chào! Tôi tên là Nam.", 4);
o console.log(test3); // CN --> Các chữ có 3 ký tự trở xuống bị bỏ,
vì n = 4
o Cách 2: Viết tắt.
o const abbreviate = (string, n = 0) => {
o return (
o string
o .split(" ")
o .filter((word) => word.length >= n)
o .map((item) => item[0])
o .join("")
o .toUpperCase()
o)
o }
o
o const test1 = abbreviate("Xin chào! Tôi tên là Nam.");
o console.log(test1); // XCTTLN
o
o const test2 = abbreviate("Xin chào! Tôi tên là Nam.", 3);
o console.log(test2); // XCTTN --> Chữ "là" bị bỏ qua, vì có 2 ký
tự, nhỏ hơn n = 3
o
o const test3 = abbreviate("Xin chào! Tôi tên là Nam.", 4);
o console.log(test3); // CN --> Các chữ có 3 ký tự trở xuống bị bỏ,
vì n = 4

```

#### **Câu 04: AlTeRnAtInG cApS (Thay đổi viết hoa, viết thường)**

- Đề bài:
  - o Tạo một hàm thay đổi kiểu viết (hoa hoặc thường) của các chữ cái trong một chuỗi.
  - o Viết xen kẽ 1 chữ hoa xong đến 1 chữ thường, lặp đi lặp lại cho đến hết chuỗi.
  - o Lưu ý:
    - Ký tự đầu tiên phải luôn luôn là HOA.
    - Bỏ qua khoảng trắng.
  - o Ví dụ:
    - "Khá Bảnh" —> "KhÁ bẢnH"
    - "Khô trước sướng sau thế mới giàu." —> "KhỔ tRưỐc SưỐnG sAu ThẾ mỎi GiÀu."
    - ""Giàu trước khổ sau thế càng đau." —> "GiÀu TrƯớC kHỔ SaU tHế CàNg ĐaU."
- Đáp án:
- const alternatingCaps = (string) => {
- let counter = 0;

```

• let array = string.split("");
•
• let newArray = array.map((character) => {
• if (character == " ") return character; // Nếu là khoảng trắng thì
in ra luôn
• counter += 1;
• return counter % 2 ? character.toUpperCase() :
character.toLowerCase(); // nếu vị trí chẵn in chữ hoa, nếu vị trí lẻ
in chữ thường.
• });
•
• newArray = newArray.join("");
• return newArray;
• };
•
• const test1 = alternatingCaps("Khá Bảnh");
• console.log(test1); // KhẤ bẢnH
•
• const test2 = alternatingCaps("Khổ trước sướng sau thế mới giàu.");
• console.log(test2); // KhỔ tRƯỚC SƯỚNG sAU ThẾ mới GiÀU.
•
• const test3 = alternatingCaps("Giàu trước khổ sau thế càng đau.");
• console.log(test3); // GiÀU TrƯỚC KHỔ SAU thẾ Càng ĐAU.

```

## Câu 05: Thống kê cơ bản: Trung vị

- Đề bài:
  - Trung vị của một nhóm số là số ở giữa khi nhóm được sắp xếp.
  - Nếu kích thước của nhóm là chẵn, trung vị là trung bình cộng của hai số ở giữa.
  - Cho một dãy số đã sắp xếp, trả về giá trị trung bình (được làm tròn đến một chữ số thập phân nếu giá trị trung bình không phải là số nguyên).
  - Ví dụ:
    - [1, 2, 4, 5, 6, 8, 8, 8, 10] —> 6
    - [2, 2, 6, 8, 8, 10, 10] —> 8
    - [1, 2, 2, 4, 7, 8, 9, 10] —> 5.5
- Đáp án:
 

```

• const sum = (arr) => {
• return arr.reduce((total, num) => total + num, 0);
• };
•
• const mean = (arr) => {
• return sum(arr) / arr.length;
• };
•
• const median = (arr) => {
• const numberBefore = arr[Math.floor((arr.length - 1) / 2)];
• const numberAfter = arr[Math.ceil((arr.length - 1) / 2)];
•
• return mean([numberBefore, numberAfter]);
• };
•
• const test1 = median([1, 2, 4, 5, 6, 8, 8, 8, 10]);
• console.log(test1); // 6
•

```

- `const test2 = median([2, 2, 6, 8, 8, 10, 10]);`
- `console.log(test2); // 8`
- 
- `const test3 = median([1, 2, 2, 4, 7, 8, 9, 10]);`
- `console.log(test3); // 5.5`

## Câu 06: Bánh sinh nhật

- Đề bài:
  - Viết một hàm **getBirthdayCake(name, age)** để tạo một chiếc bánh sinh nhật hình chữ nhật, dựa trên tên và tuổi của ai đó.
    - Nếu tuổi là một số chẵn, hãy vẽ hình bao quanh bằng chữ "#".
    - Nếu tuổi là một số lẻ, hãy vẽ hình bao quanh bằng chữ "\*".
    - Dòng tin nhắn phải ở định dạng: {tuổi} Chúc mừng sinh nhật {tên}! {tuổi}
    - Để lại một khoảng trống giữa mép bánh và số tuổi.

◦ Ví dụ:

```
◦ const test1 = getBirthdayCake("Nam", 18);
◦ console.log(test1);
◦ /*
◦ #####
◦ # 18 Chúc mừng sinh nhật Nam! 18 #
◦ #####
◦ */
◦
◦ const test2 = getBirthdayCake("Long", 17);
◦ console.log(test2);
◦ /*
◦ *****
◦ * 17 Chúc mừng sinh nhật Long! 17 *
◦ *****
◦ */
```

• Đáp án:

```
• const getBirthdayCake = (name, age) => {
• const char = age % 2 === 0 ? '#' : '*';
•
• const middle = `${char} ${age} Chúc mừng sinh nhật ${name}! ${age} ${char}`;
•
• const edge = char.repeat(middle.length); // repeat(n) để lặp lại n lần
•
• return `
• ${edge}
• ${middle}
• ${edge}
• `;
• };
•
• const test1 = getBirthdayCake("Nam", 18);
• console.log(test1);
• /*
• #####
• # 18 Chúc mừng sinh nhật Nam! 18 #
• #####
• */
```

- 
- `const test2 = getBirthdayCake("Long", 17);`
- `console.log(test2);`
- `/*`
- `*****`
- `* 17 Chúc mừng sinh nhật Long! 17 *`
- `*****`
- `*/`

## Câu 07: Blah, Blah, Blah...

- Đề bài:
  - Tạo một hàm thay thế **n** từ cuối cùng bằng "blah". Thêm "..." vào "blah" cuối cùng.
    - Nếu **n** dài hơn số từ trong câu, hãy thay tất cả các từ bằng "blah"
    - Tất cả các từ "blah" sẽ là chữ thường!
  - Ví dụ:
    - "Anh học CNTT hả, cài win dùm em nhé!" —> "Anh học CNTT hả, cài win blah blah blah..."
    - "Cảm giác bất an khi tự dừng một đứa bạn cũ hỏi mình đang học IT phải không." —> "Cảm giác bất an khi tự dừng một đứa bạn cũ hỏi mình blah blah blah blah blah..."
- Đáp án:
 

```
const blahBlah = (string, number) => {
 const words = string.split(" ");

 const newWords = words.map((word, index, arrayOrigin) => {
 return number > arrayOrigin.length - index - 1 ? "blah" : word;
 });

 const newString = newWords.join(" ");

 return `${newString}...`;
};

const test1 = blahBlah("Anh học CNTT hả, cài win dùm em nhé!", 3);
console.log(test1); // Anh học CNTT hả, cài win blah blah blah...

const test2 = blahBlah(
 "Cảm giác bất an khi tự dừng một đứa bạn cũ hỏi mình đang học IT phải không.",
 5
);
console.log(test2); // Anh học CNTT hả, cài win blah blah blah...
```

## Câu 08: Tính tổng đơn hàng

- Đề bài:
  - Tạo một hàm truyền vào một array gồm các phần tử là object (gồm các key: product, quantity, price) để tính toán tổng giá và trả về dưới dạng một số.
  - Ví dụ:
  - `const test1 = getTotalPrice([`
  - `{ product: "Sữa", quantity: 1, price: 7000 }`
  - `]);`

```

o console.log(test1); // 7000
o
o const test2 = getTotalPrice([
o { product: "Sữa", quantity: 1, price: 7000 },
o { product: "Ngũ cốc", quantity: 1, price: 50000 },
o]);
o console.log(test2); // 57000
o
o const test3 = getTotalPrice([
o { product: "Sữa", quantity: 3, price: 7000 }
o]);
o console.log(test3); // 21000
o
o const test4 = getTotalPrice([
o { product: "Sữa", quantity: 1, price: 7000 },
o { product: "Trứng", quantity: 12, price: 3000 },
o { product: "Bánh mì", quantity: 2, price: 15000 },
o { product: "Phô mai", quantity: 1, price: 5000 },
o]);
o console.log(test4); // 78000
o
o const test5 = getTotalPrice([
o { product: "Sô cô la", quantity: 1, price: 12000 },
o { product: "Kẹo", quantity: 1, price: 2000 },
o]);
o console.log(test5); // 14000

```

• **Đáp án:**

```

• const sum = (array) => {
• return array.reduce((total, num) => total + num, 0);
• };
•
• const getTotalPrice = (array) => {
• const arrayTotal = array.map((item) => item.quantity * item.price);
• return sum(arrayTotal);
• };
•
• const test1 = getTotalPrice([
• { product: "Sữa", quantity: 1, price: 7000 }
•]);
• console.log(test1); // 7000
•
• const test2 = getTotalPrice([
• { product: "Sữa", quantity: 1, price: 7000 },
• { product: "Ngũ cốc", quantity: 1, price: 50000 },
•]);
• console.log(test2); // 57000
•
• const test3 = getTotalPrice([
• { product: "Sữa", quantity: 3, price: 7000 }
•]);
• console.log(test3); // 21000
•
• const test4 = getTotalPrice([
• { product: "Sữa", quantity: 1, price: 7000 },
• { product: "Trứng", quantity: 12, price: 3000 },
• { product: "Bánh mì", quantity: 2, price: 15000 },
• { product: "Phô mai", quantity: 1, price: 5000 },
•]);
• console.log(test4); // 78000
•

```

- `const test5 = getTotalPrice([`
- `{ product: "Sô cô la", quantity: 1, price: 12000 },`
- `{ product: "Kẹo", quantity: 1, price: 2000 },`
- `]);`
- `console.log(test5); // 14000`

### Câu 09: Viết hoa chữ cái đầu tiên của mỗi từ

- Đề bài:
  - Tạo một hàm và truyền vào một chuỗi. Sau đó chuyển đổi ký tự đầu tiên của mỗi từ trong chuỗi đó thành chữ hoa. Trả về chuỗi vừa được định dạng.
  - Ví dụ:
    - "Fix bug là chuyện dễ. Tìm đoạn code gây ra bug để fix mới là chuyện khó." —> "Fix Bug Là Chuyện Dễ. Tìm Đoạn Code Gây Ra Bug Để Fix Mới Là Chuyện Khó."
    - "Khi tui biên dịch và code chạy suôn sẻ trong lần đầu. Tui tự hỏi đã làm sai chỗ nào." —> "Khi Tui Biên Dịch Và Code Chạy Suôn Sẻ Trong Lần Đầu. Tui Tự Hỏi Đã Làm Sai Chỗ Nào."
- Đáp án:
 

```
const capitalize = (string) => {
 return string.slice(0, 1).toUpperCase() + string.slice(1);
};

const makeTitle = (string) => {
 return string.split(" ").map(capitalize).join(" ");
};

const test1 = makeTitle("Fix bug là chuyện dễ. Tìm đoạn code gây ra bug để fix mới là chuyện khó.");
console.log(test1);

const test2 = makeTitle("Khi tui biên dịch và code chạy suôn sẻ trong lần đầu. Tui tự hỏi đã làm sai chỗ nào.");
console.log(test2);
```

### Câu 10: Chọn cầu chì

- Đề bài:
  - Cầu chì bị đứt mạch điện khi dòng điện đi qua vượt quá định mức của cầu chì, để ngăn không cho nhiệt tích tụ quá nhiều (có thể gây cháy). Cầu chì lý tưởng để chọn cao hơn đầu ra hiện tại của thiết bị, nhưng cũng càng gần với nó càng tốt.
  - Đưa ra một danh sách các loại cầu chì và đầu ra hiện tại của thiết bị, hãy trả về cầu chì nào là tốt nhất cho thiết bị.
  - Ví dụ:
    - Danh sách cầu chì "3V", "5V", "12V" và thiết bị "4.5V" —> "5V"
    - Danh sách cầu chì "5V", "14V", "2V" và thiết bị "5.5V" —> "14V"
    - Danh sách cầu chì "17V", "15V", "12V" và thiết bị "9V" —> "12V"
- Đáp án:
 

```
const chooseFuse = (fuses, current) => {
 const num = parseFloat(current);
```

- `const possibleFuses = fuses.map(parseFloat).filter((fuse) => fuse >= num);`
- 
- `return `${Math.min(...possibleFuses)}V`;`
- `};`
- 
- `const test1 = chooseFuse(["3V", "5V", "12V"], "4.5V");`
- `console.log(test1); // 5V`
- 
- `const test2 = chooseFuse(["5V", "14V", "2V"], "5.5V");`
- `console.log(test2); // 14V`
- 
- `const test3 = chooseFuse(["17V", "15V", "12V"], "9V");`
- `console.log(test3); // 12V`

## Câu 11: Đếm số lượng số "1" trong mảng 2D

- Đề bài:
  - Tạo hàm đếm số lượng số "1" trong mảng 2D.
  - Ví dụ:
  - `const test1 = countOnes([`
  - `[1, 0],`
  - `[0, 0],`
  - `]);`
  - `console.log(test1); // 1`
  - 
  - `const test2 = countOnes([`
  - `[1, 1, 1],`
  - `[0, 0, 1],`
  - `[1, 1, 1],`
  - `]);`
  - `console.log(test2); // 7`
  - 
  - `const test3 = countOnes([`
  - `[1, 2, 3],`
  - `[0, 2, 1],`
  - `[5, 7, 33],`
  - `]);`
  - `console.log(test3); // 2`
- Đáp án:
- `const countOnes = arr => {`
- `const result = arr.reduce((total, row) => total + row.filter(cell => cell === 1).length, 0);`
- `return result;`
- `}`
- 
- `const test1 = countOnes([`
- `[1, 0],`
- `[0, 0],`
- `]);`
- `console.log(test1); // 1`
- 
- `const test2 = countOnes([`
- `[1, 1, 1],`
- `[0, 0, 1],`
- `[1, 1, 1],`
- `]);`
- `console.log(test2); // 7`
-

- `const test3 = countOnes([`
- `[1, 2, 3],`
- `[0, 2, 1],`
- `[5, 7, 33],`
- `]);`
- `console.log(test3); // 2`

## Câu 12: Swap Cases (Đảo ngược viết hoa, viết thường)

- Đề bài:
  - Viết hàm `swappingCases(string)` truyền vào một string bất kỳ, đảo ngược cách viết hoa, viết thường của các chữ cái
  - Ví dụ:
    - `"Le VAn HunG" → "lE vaN hUNg"`
    - `"Đặng PhuƠnG NAM" → "đẶNG pHƯƠNg nAM"`
- Đáp án:
 

```
const swappingCases = (string) => {
 let swapString = string
 .split("")
 .map((char) =>
 char === char.toUpperCase() ? char.toLowerCase() :
 char.toLowerCase()
)
 .join("");

 return swapString;
};

const test1 = "Le VAn HunG";
console.log(swappingCases(test1));
// Trả về: lE vaN hUNg

const test2 = "Đặng PhuƠnG NAM";
console.log(swappingCases(test2));
// Trả về: đẶNG pHƯƠNg nAM
```

## Câu 13: Inverted Numbers (Đảo ngược giá trị của số)

- Đề bài:
  - Viết hàm `invertedNumbers(array)` truyền vào một mảng có các phần tử là các số bất kỳ, hãy đảo ngược giá trị từ âm sang dương, hoặc từ dương sang âm của các số trong mảng đó.
  - Ví dụ:
    - `[1, -10, -20, 15, 100, -30] → [-1, 10, 20, -15, -100, 30]`
    - `[-20, 30, 10, -25, -60, 20] → [20, -30, -10, 25, 60, -20]`
- Đáp án:
 

```
const invertedNumbers = (array) => {
 const newArray = array.map((item) => -item);
 return newArray;
};

const test1 = [1, -10, -20, 15, 100, -30];
console.log(invertedNumbers(test1));
```

- // Trả về: [-1, 10, 20, -15, -100, 30];
- 
- const test2 = [-20, 30, 10, -25, -60, 20];
- console.log(invertedNumbers(test2));
- // Trả về: [20, -30, -10, 25, 60, -20];

#### Câu 14: Ignore Number Value (Bỏ qua các chữ số)

- Đề bài:
  - Viết hàm ignoreNumbers(string) truyền vào một string bao gồm cả số và chữ, hãy trả về một string mới chỉ gồm chữ.
  - Ví dụ:
    - "Test4Ag54SF" → "TestAgSF"
    - "JHk34Gl3gG" → "JHkGlgG"
- Đáp án:
 

```
const ignoreNumbers = (string) => {
 const newString = string
 .split("")
 .map((item) => (isNaN(parseInt(item)) ? item : ""))
 .join("");
 return newString;
};

const test1 = "Test4Ag54SF";
console.log(ignoreNumbers(test1));
// Trả về: TestAgSF

const test2 = "JHk34Gl3gG";
console.log(ignoreNumbers(test2));
// Trả về: JHkGlgG
```

#### Câu 15: Tìm từ ≤ n ký tự

- Đề bài:
  - Viết hàm smallWords(string, number) truyền vào một string và một số number. Trả ra một string mới chỉ gồm những từ ≤ number.
  - Ví dụ:
    - smallWords("I Love Foood Code Too Playing Much", 4) → "I Love Code Too Much".
    - smallWords("I Love Foood Code Too Playing Much", 3) → "I Too"
- Đáp án:
 

```
const smallWords = (string, number) => {
 const newString = string
 .split(" ")
 .filter((item) => item.length <= number)
 .join(" ");
 return newString;
};

const test1 = "I Love Foood Code Too Playing Much";
console.log(smallWords(test1, 4));
// Trả về: I Love Code Too Much
```

- `const test2 = "I Love Foood Code Too Playing Much";`
- `console.log(smallWords(test2, 3));`
- `// Trả về: I Too`

## Câu 16: Tìm số trong String và Nhân

- Đề bài:
  - Viết hàm `multiplyNumberInString(string)`, truyền vào một string, hãy trả ra một string mới là phép nhân của các số có trong string đó.
  - Ví dụ:
    - `"JG23BGH5BA" —> "4925"`
      - $2*2 = 4$
      - $3*3 = 9$
      - $5*5 = 25$
      - $\rightarrow "4925"$
    - `"VD23GS8S6AH" —> "496436"`
      - $2*2 = 4$
      - $3*3 = 9$
      - $8*8 = 64$
      - $6*6 = 36$
      - $\rightarrow "496436"$
    - `"AGD353GDSK8" —> "925964"`
      - $3*3 = 9$
      - $5*5 = 25$
      - $3*3 = 9$
      - $8*8 = 64$
      - $\rightarrow "925964"$
    - `"JBKJJKLDJ" —> "0"`
      - Do không có số nào.

- Đáp án:
 

```
const multiplyNumberInString = (string) => {
 const result = string
 .split("")
 .filter((item) => !isNaN(parseInt(item)))
 .map((item) => item * item)
 .join("");
 return result.length > 0 ? result : "0";
};

const test1 = "JG23BGH5BA";
console.log(multiplyNumberInString(test1)); // "4925"

const test2 = "VD23GS8S6AH";
console.log(multiplyNumberInString(test2)); // "496436"
```

- `const test3 = "AGD353GDSK8";`
- `console.log(multiplyNumberInString(test3)); // "925964"`
- 
- `const test4 = "JBKJJKLJDJ";`
- `console.log(multiplyNumberInString(test4)); // "0"`

## Câu 17: Tính tổng tiền trong giỏ hàng

- Đề bài:
  - Cho một mảng giỏ hàng chứa các sản phẩm (bao gồm tên, giá, số lượng). Hãy tính tổng tiền của giỏ hàng đó.
  - Ví dụ:
  - `const cart = [`
  - `{ name: "iPhone", price: 1000, quantity: 5 },`
  - `{ name: "iPad", price: 500, quantity: 2 },`
  - `{ name: "MacBook", price: 2000, quantity: 1 },`
  - `];`
  - 
  - `// Tính toán để trả về: 1000 * 5 + 500 * 2 + 2000 * 1 = 8000`
- Đáp án:
- `const cart = [`
- `{ name: "iPhone", price: 1000, quantity: 5 },`
- `{ name: "iPad", price: 500, quantity: 2 },`
- `{ name: "MacBook", price: 2000, quantity: 1 },`
- `];`
- 
- `// Tính toán để trả về: 1000 * 5 + 500 * 2 + 2000 * 1 = 8000`
- 
- `const totalPrice = cart.reduce(`
- `(total, item) => total + item.price * item.quantity,`
- `0`
- `);`
- 
- `console.log(totalPrice); // Trả về: 8000`

## Câu 18: Nhóm các học sinh trong một lớp theo giới tính

- Đề bài:
  - Cho một mảng danh sách các học sinh của một lớp. Hãy nhóm các học sinh **Nam** thành 1 nhóm, học sinh **Nữ** thành 1 nhóm.
  - Ví dụ:
  - `const students = [`
  - `{ hoTen: "Le Van A", gioiTinh: "Nam" },`
  - `{ hoTen: "Do Van B", gioiTinh: "Nam" },`
  - `{ hoTen: "Nguyen Thi C", gioiTinh: "Nu" },`
  - `{ hoTen: "Dao Van D", gioiTinh: "Nam" },`
  - `{ hoTen: "Hoang Thi E", gioiTinh: "Nu" },`
  - `{ hoTen: "Vu Van F", gioiTinh: "Nam" },`
  - `];`
  - 
  - `/*`
  - `Trả về:`
  - `{`
  - `"Nam": [`
  - `"Le Van A",`
  - `"Do Van B",`
  - `"Dao Van D",`
  - `"Vu Van F"`

```

 ○],
 ○ "Nu": [
 ○ "Nguyen Thi C",
 ○ "Hoang Thi E"
 ○]
 ○ }
 ○ */

```

- **Đáp án:**

```

• const students = [
• { hoTen: "Le Van A", gioiTinh: "Nam" },
• { hoTen: "Do Van B", gioiTinh: "Nam" },
• { hoTen: "Nguyen Thi C", gioiTinh: "Nu" },
• { hoTen: "Dao Van D", gioiTinh: "Nam" },
• { hoTen: "Hoang Thi E", gioiTinh: "Nu" },
• { hoTen: "Vu Van F", gioiTinh: "Nam" },
•];
•
• const groupedStudents = students.reduce((groups, item) => {
• if (groups[item.gioiTinh]) {
• groups[item.gioiTinh].push(item.hoTen);
• } else {
• groups[item.gioiTinh] = [item.hoTen];
• }
• return groups;
• }, {});
•
• console.log(groupedStudents);
• /*
• Trả về :
• {
• "Nam": [
• "Le Van A",
• "Do Van B",
• "Dao Van D",
• "Vu Van F"
•],
• "Nu": [
• "Nguyen Thi C",
• "Hoang Thi E"
•]
• }
• */

```

## Câu 19: Tính tổng theo từng nhóm trong một mảng

- **Đề bài:**

- Cho một mảng gồm các phần tử là các mảng con. Hãy tính tổng giá trị của các phần tử trong mảng con.
- Ví dụ:
- ```
const numbers = [
  [1, 2],
  [3, 4],
  [5, 6],
];
```
- ```
/*
 Trả về :
 [3, 7, 11];
 */
```

- Đáp án:
- ```
const numbers = [
  [1, 2],
  [3, 4],
  [5, 6],
];

const sumByGroup = numbers.reduce((groups, item) => {
  const sum = item.reduce((total, value) => total + value, 0);
  groups.push(sum);
  return groups;
}, []);

console.log(sumByGroup);
/*
Trả về:
[3, 7, 11];
*/
```

Câu 20: Tính tổng điểm theo từng lớp

- Đề bài:
 - Cho một mảng gồm các học sinh (bao gồm các thông tin: họ tên, lớp, điểm trung bình của học sinh đó). Hãy tính tổng điểm theo từng lớp.
 - Ví dụ:
 - ```
const students = [
 { hoTen: "Le Van A", lop: "A", diem: 7.5 },
 { hoTen: "Do Van B", lop: "B", diem: 6.8 },
 { hoTen: "Nguyen Thi C", lop: "A", diem: 8 },
 { hoTen: "Dao Van D", lop: "C", diem: 9 },
 { hoTen: "Hoang Thi E", lop: "B", diem: 8.5 },
 { hoTen: "Vu Van F", lop: "B", diem: 7.2 },
];

/*
Trả về:
{
 A: 15.5,
 B: 22.5,
 C: 9
}
*/
```
- Đáp án:
- ```
const students = [
  { hoTen: "Le Van A", lop: "A", diem: 7.5 },
  { hoTen: "Do Van B", lop: "B", diem: 6.8 },
  { hoTen: "Nguyen Thi C", lop: "A", diem: 8 },
  { hoTen: "Dao Van D", lop: "C", diem: 9 },
  { hoTen: "Hoang Thi E", lop: "B", diem: 8.5 },
  { hoTen: "Vu Van F", lop: "B", diem: 7.2 },
];

const sumByClass = students.reduce((totalByClass, item) => {
  const lop = item.lop;
  if (!totalByClass[lop]) {
    totalByClass[lop] = 0;
```

- }
- totalByClass[lop] += item.diem;
- return totalByClass;
- }, {});
-
- console.log(sumByClass);
- /*
- Trả về:
- {
- A: 15.5,
- B: 22.5,
- C: 9
- }
- */
-