

# Tổng hợp các bài tập lập trình cơ bản

---



## Nguồn tham khảo:

- [1000 bài tập của thầy Khang](#)
- [Edabit](#)
- [LeetCode](#) cùng một số nguồn tham khảo khác.

## AGENDA

1. Cách làm bài tập trong file pdf này
2. Xử lý số (number)
3. Xử lý chuỗi (string)
4. Xử lý mảng (array): kiểm tra phần tử có trong mảng không?
5. Xử lý mảng (array): tìm kiếm phần tử
6. Xử lý mảng (array): liệt kê
7. Xử lý mảng (array): đếm
8. Xử lý mảng (array): tính tổng
9. Xử lý mảng (array): thêm / xoá phần tử
10. Xử lý mảng (array): mảng con
11. Bài tập tổng hợp luyện tư duy

# DANH MỤC BÀI TẬP

- Tổng hợp các bài tập lập trình cơ bản
  - 1. Cách làm bài tập trong file pdf này
  - 2. Xử lý số (number)
    - **NUMBER-01:** Kiểm tra số tăng dần
    - **NUMBER-02:** Kiểm tra số giảm dần
    - **NUMBER-03:** Kiểm tra số tăng dần đều theo khoảng cách cho trước
    - **NUMBER-04:** Kiểm tra số nguyên tố
    - **NUMBER-05:** Kiểm tra số chính phương
    - **NUMBER-06:** Kiểm tra số hoàn hảo
    - **NUMBER-07:** Kiểm tra số đối xứng
    - **NUMBER-08:** Kiểm tra số có tổng chữ số chia hết cho 10
    - **NUMBER-09:** Kiểm tra số có tổng của 2 chữ số bằng số cho trước
  - 3. Xử lý chuỗi (string)
    - **STRING-01:** Đếm số từ có trong câu
    - **STRING-02:** Thống kê số lượng từ trong câu
    - **STRING-03:** Thống kê ký tự có trong câu
    - **STRING-04:** Đếm số lượng email có trong câu
    - **STRING-05:** Đếm số lượng url có trong câu
    - **STRING-06:** Hiển thị địa chỉ người dùng
    - **STRING-07:** Thay thế params thành value tương ứng
  - 4. Xử lý mảng (array): kiểm tra phần tử có trong mảng không?
    - **ARRAY-CHECK-01:** Kiểm tra có số chẵn lớn hơn n trong mảng không
    - **ARRAY-CHECK-02:** Kiểm tra có số lẻ và chia hết cho 3 không
    - **ARRAY-CHECK-03:** Kiểm tra có từ easy frontend trong mảng không
    - **ARRAY-CHECK-04:** Kiểm tra có giá trị truthy nào trong mảng không
    - **ARRAY-CHECK-05:** Kiểm tra có giá trị falsy nào trong mảng không
    - **ARRAY-CHECK-06:** Kiểm tra có student có id = 123 trong mảng không
    - **ARRAY-CHECK-07:** Kiểm tra có student giới tính là Nữ tên Alice không
    - **ARRAY-CHECK-08:** Kiểm tra có sản phẩm lớn hơn 100000 và free ship không
    - **ARRAY-CHECK-09:** Kiểm tra mảng có tăng dần không
    - **ARRAY-CHECK-10:** Kiểm tra mảng có giảm dần không
    - **ARRAY-CHECK-11:** Kiểm tra mảng có đối xứng không
    - **ARRAY-CHECK-12:** Kiểm tra mảng có số fibonacci nhỏ hơn 100 không?
  - 5. Xử lý mảng (array): tìm kiếm phần tử
    - **ARRAY-FIND-01:** Tìm số dương nhỏ nhất trong mảng
    - **ARRAY-FIND-02:** Tìm từ dài nhất nhưng nhỏ hơn 5 trong mảng
    - **ARRAY-FIND-03:** Tìm số dương chẵn đầu tiên trong mảng
    - **ARRAY-FIND-04:** Tìm số âm lẻ cuối cùng trong mảng
    - **ARRAY-FIND-05:** Tìm số lớn thứ 2 trong mảng
    - **ARRAY-FIND-06:** Tìm số chính phương cuối cùng trong mảng
    - **ARRAY-FIND-07:** Tìm email đầu tiên trong mảng
    - **ARRAY-FIND-08:** Tìm url cuối cùng trong mảng

- **ARRAY-FIND-09:** Tìm vị trí của student có id = 123
- **ARRAY-FIND-10:** Tìm vị trí của sản phẩm có mã code là = abc
- **ARRAY-FIND-11:** Tìm student có điểm môn toán lớn nhất đầu tiên
- **ARRAY-FIND-12:** Tìm student có điểm trung bình môn thấp nhất cuối cùng
- **ARRAY-FIND-13:** Tìm student đầu tiên có tất cả điểm trên 5, nhưng bị lại một môn dưới 5
- **ARRAY-FIND-14:** Tìm student có tổng điểm cao nhất nhưng đã nghỉ học, trường hợp có nhiều bạn điểm bằng nhau thì trả về bạn đầu tiên
- **ARRAY-FIND-15:** Tìm sản phẩm đầu tiên có free ship
- **ARRAY-FIND-16:** Tìm sản phẩm có giá cao nhất trong nhóm sản phẩm điện thoại
- **ARRAY-FIND-17:** Tìm sản phẩm đầu tiên có tên bắt đầu bằng chữ iphone (không phân biệt hoa thường)
- 6. Xử lý mảng (array): liệt kê
  - **ARRAY-FILTER-01:** Liệt kê các số trong khoảng [a, b]
  - **ARRAY-FILTER-02:** Tìm tất cả các số mà nó lớn hơn số đứng trước nó
  - **ARRAY-FILTER-03:** Dãy số fibonacci nhỏ hơn n
  - **ARRAY-FILTER-04:** Dãy số nguyên tố nhỏ hơn n
  - **ARRAY-FILTER-05:** Tìm tất cả các số bằng với số dương chẵn đầu tiên trong mảng
  - **ARRAY-FILTER-06:** Tìm tất cả các số mà bắt đầu bằng chữ số lẻ
  - **ARRAY-FILTER-07:** Tìm tất cả student có điểm toán nhỏ hơn 5
  - **ARRAY-FILTER-08:** Tìm tất cả sản phẩm có hỗ trợ free ship
  - **ARRAY-FILTER-09:** Tìm tất cả sản phẩm thuộc dòng iphone có giá nhỏ hơn 10tr
- 7. Xử lý mảng (array): đếm
  - **ARRAY-COUNT-01:** Đếm số dương chẵn có trong mảng
  - **ARRAY-COUNT-02:** Đếm số mà nó nhỏ hơn số liền trước của nó
  - **ARRAY-COUNT-03:** Đếm số từ có độ dài lớn hơn 5 có trong mảng
  - **ARRAY-COUNT-04:** Đếm số lượng URLs có trong mảng
  - **ARRAY-COUNT-05:** Đếm số lượng các số khác nhau có trong mảng
  - **ARRAY-COUNT-06:** Đếm số lượng các số có trong mảng a mà không có trong mảng b
  - **ARRAY-COUNT-07:** Đếm số lượng students có điểm trung bình lớn hơn 7.5
  - **ARRAY-COUNT-08:** Đếm số lượng students lớp 12
  - **ARRAY-COUNT-09:** Đếm số lượng sản phẩm thuộc dòng iphone và có hỗ trợ free ship
  - **ARRAY-COUNT-10:** Đếm số lượng sản phẩm có trạng thái là available
- 8. Xử lý mảng (array): tính tổng
  - **ARRAY-SUM-01:** Tổng các số chẵn mà nó lớn hơn số trước đó.
  - **ARRAY-SUM-02:** Tổng của các chữ số trong mảng
  - **ARRAY-SUM-03:** Tổng các chữ số nhỏ nhất của số trong mảng
  - **ARRAY-SUM-04:** Tổng điểm toán của các students có gender là male
  - **ARRAY-SUM-05:** Tổng tiền của giỏ hàng
- 9. Xử lý mảng (array): thêm / xóa phần tử
  - **ARRAY-CD-01:** Thêm một phần tử tại vị trí k
  - **ARRAY-CD-02:** Xóa một phần tử tại vị trí k
  - **ARRAY-CD-03:** Xóa một student có id cho trước
  - **ARRAY-CD-04:** Cho một mảng đã được sắp xếp tăng dần. Cho một số, nhờ bạn add vào mảng nhưng phải đảm bảo mảng vẫn đc sắp xếp tăng dần.
  - **ARRAY-CD-05:** Xóa các số chẵn trong mảng

- **ARRAY-CD-06:** Xoá các số trùng nhau trong mảng
- **ARRAY-CD-07:** Xoá các số trùng nhau nhưng giữ lại một số
- 10. Xử lý mảng (array): mảng con
  - **ARRAY-SUBARR-01:** Tìm các mảng con tăng dần
  - **ARRAY-SUBARR-02:** Tìm các mảng con giảm dần có ít nhất 2 phần tử
  - **ARRAY-SUBARR-03:** Kiểm tra mảng a có phải là mảng con của mảng b
  - **ARRAY-SUBARR-04:** Tìm các mảng con có chứa số dương chẵn liên tiếp
  - **ARRAY-SUBARR-05:** Tìm mảng con có tổng lớn nhất, trả về con số tổng
  - **ARRAY-SUBARR-06:** Tìm mảng con có tổng lớn nhất, trả về mảng con đầu tiên
  - **ARRAY-SUBARR-07:** Tìm mảng con có số âm dương xen kẽ dài nhất
- 11. Bài tập tổng hợp luyện tư duy
  - **LOGIC-01:** Tìm 2 số có tổng bằng số cho trước
  - **LOGIC-02:** Tìm số trong mảng cách xa với số cho trước nhất
  - **LOGIC-03:** Tìm số bị thiếu cho trong một mảng từ [5..n]
  - **LOGIC-04:** Thống kê số lần xuất hiện của các số trong mảng
  - **LOGIC-05:** Tìm số có lần xuất hiện nhiều nhất
  - **LOGIC-06:** Tìm ký tự xuất hiện nhiều nhất trong câu (trừ khoảng trắng)

## 1. Cách làm bài tập trong file pdf này

#	Step	Desc
1	Pick an exercise	Chọn một bài tập
2	Create two files	Tạo một file <code>.js</code> để code và một file <code>.test.js</code> để viết unit test. Hoặc lười thì viết code vào file unit test luôn 😊
3	Write unit test	Đọc hiểu function và xác định mong muốn, từ đó định nghĩa các test cases tương ứng
5	Coding	Viết code để cover tất cả test cases
4	Check unit test + fix bugs	Sau khi code xong thì check unit test, nếu pass hết thì okie, ko thì code tiếp 😊

Cần lưu ý là phải handle cả những trường hợp tham số truyền vào không hợp lệ nhé, đừng chỉ làm happy cases nhen hehe

Lưu ý: Với mỗi bài tập, hãy cố gắng viết bằng nhiều cách nhất có thể.

Để dễ dàng trao đổi và tham chiếu, mình sẽ đặt mã số cho mỗi bài tập:

- **NUMBER-X**: bài tập về số
- **STRING-X**: bài tập về chuỗi
- **ARRAY-CATEGORY-X**: bài tập về mảng
- **LOGIC-X**: bài tập tổng hợp luyện logic

Trong đó:

- **CATEGORY** là các chủ đề khác nhau trong xử lý mảng.
- **X** là số thứ tự 1, 2, 3, ...

## 2. Xử lý số (number)

### NUMBER-01: Kiểm tra số tăng dần

Viết hàm **isIncreasingNumber(n)** để kiểm tra **n** có phải là số tăng dần hay không?

- $0 < n < 1000000$
- Số tăng dần có ít nhất 2 chữ số
- Chữ số bên phải luôn lớn hơn chữ số bên trái

Trả về true nếu là số tăng dần, ngược lại trả về false.

**Ví dụ:**

- `isIncreasingNumber(11) --> false`
  - `isIncreasingNumber(123) --> true`
  - `isIncreasingNumber(12321) --> false`
- 

### NUMBER-02: Kiểm tra số giảm dần

Viết hàm **isDecreasingNumber(n)** để kiểm tra **n** có phải là số giảm dần hay không?

- $0 < n < 1000000$
- Số giảm dần có ít nhất 2 chữ số
- Chữ số bên phải luôn nhỏ hơn chữ số bên trái

Trả về true nếu là số giảm dần, ngược lại trả về false.

**Ví dụ:**

- `isDecreasingNumber(11) --> false`
  - `isDecreasingNumber(321) --> true`
  - `isDecreasingNumber(12321) --> false`
-

### NUMBER-03: Kiểm tra số tăng dần đều theo khoảng cách cho trước

Viết hàm **isIncreasingNumberByDistance(n, x)** để kiểm tra **n** có phải là số tăng dần đều với khoảng cách giữa 2 chữ số là **x** không?

- $0 < n < 1000000$ ,  $0 < x < 5$
- Số tăng dần có ít nhất 2 chữ số
- Chữ số bên phải luôn lớn hơn chữ số bên trái
- Khoảng cách của 2 chữ số liền kề là **x**

Trả về true nếu là số tăng dần đều theo khoảng cách x, ngược lại trả về false.

Ví dụ:

- `isIncreasingNumberByDistance(11, 1) --> false`
  - `isIncreasingNumberByDistance(123, 1) --> true`
  - `isIncreasingNumberByDistance(135, 2) --> true`
  - `isIncreasingNumberByDistance(135, 1) --> false`
- 

### NUMBER-04: Kiểm tra số nguyên tố

Viết hàm **isPrime(n)** để kiểm tra n có phải là số nguyên tố không?

- Số nguyên tố là số chỉ chia hết cho 1 và chính nó.
- $0 < n < 100000$

Trả về true nếu là số nguyên tố, ngược lại trả về false.

Ví dụ:

- `isPrime(3) --> true`
  - `isPrime(4) --> false`
  - `isPrime(11) --> true`
- 

### NUMBER-05: Kiểm tra số chính phương

Viết hàm **isPerfectSquare(n)** để kiểm tra n có phải là số chính phương không?

- $0 < n < 100000$
- Số chính phương là số có căn bậc 2 của nó là một số tự nhiên.
- Trả về true nếu là số chính phương, ngược lại trả về false.

Ví dụ: 4, 9, 25, 36 là các số chính phương.

---

## NUMBER-06: Kiểm tra số hoàn hảo

Viết hàm **isPerfectNumber(n)** để kiểm tra n có phải là số hoàn hảo hay không?

- Với n thỏa điều kiện  $1 < n < 1000$
- Số hoàn hảo là số mà tổng của tất cả ước số (không tính chính nó, tức từ 1 đến  $n - 1$ ) bằng chính nó.
- Trả về true nếu đúng, ngược lại trả về false

**Ví dụ:**  $6 = 1 + 2 + 3$  (như vậy 6 là một số hoàn hảo)

Gợi ý: không nhất thiết phải chạy tới  $(n - 1)$  để tìm ra tất cả các ước số của n

---

## NUMBER-07: Kiểm tra số đối xứng

Viết hàm **isSymetricNumber(n)** để kiểm tra n có phải là số đối xứng hay không?

- Với n thỏa điều kiện  $1 < n < 1000000$
- Số đối xứng là số đọc từ trái sang phải cũng giống như đọc từ phải sang trái.
- Trả về true nếu đúng, ngược lại trả về false

**Ví dụ:**

- `isSymetricNumber(10) --> false`
- `isSymetricNumber(11) --> true`
- `isSymetricNumber(12321) --> true`

---

## NUMBER-08: Kiểm tra số có tổng chữ số chia hết cho 10

Viết hàm **isDivisibleBy10(n)** để kiểm tra n ( $0 < n < 1000000$ ) có tổng chữ số chia hết cho 10 hay không?

**Ví dụ:**

- `isDivisibleBy10(10) --> false`
- `isDivisibleBy10(1234) --> true` vì  $1 + 2 + 3 + 4 = 10$  chia hết cho 10
- `isDivisibleBy10(123455) --> true` vì  $1 + 2 + 3 + 4 + 5 + 5 = 20$  chia hết cho 10

---

## NUMBER-09: Kiểm tra số có tổng của 2 chữ số bằng số cho trước

Viết hàm **hasTwoDigitsWithSum(n, sum)** để kiểm tra n ( $9 < n < 1000000$ ) có chứa **2 chữ số bất kỳ** nào có tổng bằng **sum** ( $0 < \text{sum} < 19$ ) cho trước không?

**Ví dụ:**

- `hasTwoDigitsWithSum(10, 1) --> true` vì  $0 + 1 = 1$
  - `hasTwoDigitsWithSum(101, 3) --> false` vì không có 2 chữ số nào cộng lại bằng 3
  - `hasTwoDigitsWithSum(11, 2) --> true`
-



### 3. Xử lý chuỗi (string)

#### STRING-01: Đếm số từ có trong câu

Viết hàm **countWords(str)** để đếm số từ có trong **str**.

**Ví dụ:** countWords('easy frontend') --> 2

---

#### STRING-02: Thống kê số lượng từ trong câu

Viết hàm **statisticsWords(str)** để đếm tần số xuất hiện của mỗi từ trong **str**.

Trả về **một object** với:

- **key** là từ có xuất hiện trong **str**
- **value** là số lần xuất hiện của **key**

**Ví dụ:**

```
statisticsWords('easy frontend easy')  
// should return an object like this:  
// { easy: 2, frontend: 1 }
```

---

#### STRING-03: Thống kê ký tự có trong câu

Viết hàm **statisticsCharacters(str)** để thống kê số lượng ký tự có trong câu.

Trả về **một object** với:

- **key** là character có xuất hiện trong **str**. Nếu là khoảng trắng thì **key**="space".
- **value** là số lần xuất hiện của **key**

**Ví dụ:**

```
statisticsCharacters('aa b cc ')  
// should return an object like this:  
// { a: 2, b: 1, c: 2, space: 3 }
```

---

## STRING-04: Đếm số lượng email có trong câu

Viết hàm **countEmails(str)** để đếm số lượng email có trong **str**

- Email là một chuỗi có chứa ký tự @ ở giữa và không phải ở cuối từ.
- Email có thể có một số domain như: **.com**, **.vn**, **.com.vn**. Domain sẽ xuất hiện ở cuối từ.
- Email có định dạng là **X@Y**. Trong đó **X** có độ dài ít nhất 3 ký tự, **Y** có độ dài ít nhất 3 ký tự (ko tính phần domain nêu trên)

Trả về số lượng email tìm thấy từ chuỗi truyền vào.

Lưu ý: không sử dụng regular expression.

### Ví dụ:

- countEmails('my email should be abc@super.com') --> 1
- countEmails('my email should be a@a.com') --> 0
- countEmails('my email should be @ bla .com') --> 0
- countEmails('my email should be easy@frontend.com or easy@frontend.com.vn or easy@frontend.vn') --> 3

---

## STRING-05: Đếm số lượng url có trong câu

Viết hàm **countURLs(str)** để đếm số lượng URLs có trong câu.

- URL có định dạng: protocol://domain.com/path-name
- URL có thể bắt đầu bằng protocol như: http, https, ws hoặc wss
- Domain có thể là .com, .com.vn hoặc .vn
- Phần tên domain phải có ít nhất 3 ký tự, như abc.com, chứ **ab.com** là không hợp lệ

Trả về số lượng URLs tìm thấy trong câu.

Lưu ý: không sử dụng regular expression.

### Ví dụ:

- countURLs('my website is: http://ezfrontend.com you can visit it') --> 1
- countURLs('my website is: https://ez.com you can visit it') --> 0 vì phần tên domain chỉ có 2 ký tự **ez** nên không hợp lệ

## STRING-06: Hiển thị địa chỉ người dùng

Viết hàm **getDisplayedAddress(address)** để ghép chuỗi từ các thuộc tính của object **address** thành một chuỗi address hoàn chỉnh

- Chuỗi trả về có dạng: "address.number address.street, address.ward, address.district, address.city"
- Lưu ý object **address** không phải lúc nào cũng có đầy đủ các thông tin trên.

Ví dụ:

```
getDisplayedAddress({
  number: 123,
  street: 'Nguyen Cong Tru',
  ward: 'Ward 11',
  district: 'Binh Thanh District',
  city: 'HCMC'
});
// should return '123 Nguyen Cong Tru, Ward 11, Binh Thanh District, HCMC'

getDisplayedAddress({
  street: 'Nguyen Cong Tru',
  district: 'Binh Thanh District'
});
// should return 'Nguyen Cong Tru, Binh Thanh District'
```

Lưu ý: để ý dấu phẩy và khoảng trắng.

## STRING-07: Thay thế params thành value tương ứng

Viết hàm **fillPath(path, params)** để thay thế các chuỗi params trong **path** bằng các giá trị tương ứng trong object **params**.

Ví dụ:

```
fillPath('/products/:productId', { productId: 123 }); // '/products/123'

fillPath('/categories/:categoryId/products/:productId', {
  categoryId: 1,
  productId: 2
});
// should return '/categories/1/products/2'

fillPath('/categories/:categoryId/products/:productId', { productId: 2 });
// should return '/categories/:categoryId/products/2'
```

## 4. Xử lý mảng (array): kiểm tra phần tử có trong mảng không?

---

**ARRAY-CHECK-01:** Kiểm tra có số chẵn lớn hơn n trong mảng không

---

**ARRAY-CHECK-02:** Kiểm tra có số lẻ và chia hết cho 3 không

---

**ARRAY-CHECK-03:** Kiểm tra có từ easy frontend trong mảng không

---

**ARRAY-CHECK-04:** Kiểm tra có giá trị truthy nào trong mảng không

---

**ARRAY-CHECK-05:** Kiểm tra có giá trị falsy nào trong mảng không

---

**ARRAY-CHECK-06:** Kiểm tra có student có id = 123 trong mảng không

---

**ARRAY-CHECK-07:** Kiểm tra có student giới tính là Nữ tên Alice không

---

**ARRAY-CHECK-08:** Kiểm tra có sản phẩm lớn hơn 100000 và free ship không

---

**ARRAY-CHECK-09:** Kiểm tra mảng có tăng dần không

---

**ARRAY-CHECK-10:** Kiểm tra mảng có giảm dần không

---

**ARRAY-CHECK-11:** Kiểm tra mảng có đối xứng không

---

**ARRAY-CHECK-12:** Kiểm tra mảng có số fibonacci nhỏ hơn 100 không?

## 5. Xử lý mảng (array): tìm kiếm phần tử

**ARRAY-FIND-01:** Tìm số dương nhỏ nhất trong mảng

---

**ARRAY-FIND-02:** Tìm từ dài nhất nhưng nhỏ hơn 5 trong mảng

---

**ARRAY-FIND-03:** Tìm số dương chẵn đầu tiên trong mảng

---

**ARRAY-FIND-04:** Tìm số âm lẻ cuối cùng trong mảng

---

**ARRAY-FIND-05:** Tìm số lớn thứ 2 trong mảng

---

**ARRAY-FIND-06:** Tìm số chính phương cuối cùng trong mảng

---

**ARRAY-FIND-07:** Tìm email đầu tiên trong mảng

---

**ARRAY-FIND-08:** Tìm url cuối cùng trong mảng

---

**ARRAY-FIND-09:** Tìm vị trí của student có id = 123

---

**ARRAY-FIND-10:** Tìm vị trí của sản phẩm có mã code là = abc

---

**ARRAY-FIND-11:** Tìm student có điểm môn toán lớn nhất đầu tiên

---

**ARRAY-FIND-12:** Tìm student có điểm trung bình môn thấp nhất cuối cùng

---

**ARRAY-FIND-13:** Tìm student đầu tiên có tất cả điểm trên 5, nhưng bị lại một môn dưới 5

---

**ARRAY-FIND-14:** Tìm student có tổng điểm cao nhất nhưng đã nghỉ học, trường hợp có nhiều bạn điểm bằng nhau thì trả về bạn đầu tiên

---

**ARRAY-FIND-15:** Tìm sản phẩm đầu tiên có free ship

---

**ARRAY-FIND-16:** Tìm sản phẩm có giá cao nhất trong nhóm sản phẩm điện thoại

---

**ARRAY-FIND-17:** Tìm sản phẩm đầu tiên có tên bắt đầu bằng chữ iphone (không phân biệt hoa thường)

## 6. Xử lý mảng (array): liệt kê

---

**ARRAY-FILTER-01:** Liệt kê các số trong khoảng [a, b]

**ARRAY-FILTER-02:** Tìm tất cả các số mà nó lớn hơn số đứng trước nó

---

**ARRAY-FILTER-03:** Dãy số fibonacci nhỏ hơn n

---

**ARRAY-FILTER-04:** Dãy số nguyên tố nhỏ hơn n

---

**ARRAY-FILTER-05:** Tìm tất cả các số bằng với số dương chẵn đầu tiên trong mảng

---

**ARRAY-FILTER-06:** Tìm tất cả các số mà bắt đầu bằng chữ số lẻ

---

**ARRAY-FILTER-07:** Tìm tất cả student có điểm toán nhỏ hơn 5

---

**ARRAY-FILTER-08:** Tìm tất cả sản phẩm có hỗ trợ free ship

---

**ARRAY-FILTER-09:** Tìm tất cả sản phẩm thuộc dòng iphone có giá nhỏ hơn 10tr

## 7. Xử lý mảng (array): đếm

**ARRAY-COUNT-01:** Đếm số dương chẵn có trong mảng

---

**ARRAY-COUNT-02:** Đếm số mà nó nhỏ hơn số liền trước của nó

---

**ARRAY-COUNT-03:** Đếm số từ có độ dài lớn hơn 5 có trong mảng

---

**ARRAY-COUNT-04:** Đếm số lượng URLs có trong mảng

---

**ARRAY-COUNT-05:** Đếm số lượng các số khác nhau có trong mảng

---

**ARRAY-COUNT-06:** Đếm số lượng các số có trong mảng a mà không có trong mảng b

---

**ARRAY-COUNT-07:** Đếm số lượng students có điểm trung bình lớn hơn 7.5

---

**ARRAY-COUNT-08:** Đếm số lượng students lớp 12

---

**ARRAY-COUNT-09:** Đếm số lượng sản phẩm thuộc dòng iphone và có hỗ trợ free ship

---

**ARRAY-COUNT-10:** Đếm số lượng sản phẩm có trạng thái là available

## 8. Xử lý mảng (array): tính tổng

**ARRAY-SUM-01:** Tổng các số chẵn mà nó lớn hơn số trước đó.

---

**ARRAY-SUM-02:** Tổng của các chữ số trong mảng

---

**ARRAY-SUM-03:** Tổng các chữ số nhỏ nhất của số trong mảng

---

**ARRAY-SUM-04:** Tổng điểm toán của các students có gender là male

---

**ARRAY-SUM-05:** Tổng tiền của giỏ hàng



## 9. Xử lý mảng (array): thêm / xoá phần tử

---

**ARRAY-CD-01:** Thêm một phần tử tại vị trí k

---

**ARRAY-CD-02:** Xoá một phần tử tại vị trí k

---

**ARRAY-CD-03:** Xoá một student có id cho trước

---

**ARRAY-CD-04:** Cho một mảng đã được sắp xếp tăng dần. Cho một số, nhờ bạn add vào mảng nhưng phải đảm bảo mảng vẫn đc sắp xếp tăng dần.

---

**ARRAY-CD-05:** Xoá các số chẵn trong mảng

---

**ARRAY-CD-06:** Xoá các số trùng nhau trong mảng

---

**ARRAY-CD-07:** Xoá các số trùng nhau nhưng giữ lại một số

## 10. Xử lý mảng (array): mảng con

---

**ARRAY-SUBARR-01:** Tìm các mảng con tăng dần

---

**ARRAY-SUBARR-02:** Tìm các mảng con giảm dần có ít nhất 2 phần tử

---

**ARRAY-SUBARR-03:** Kiểm tra mảng a có phải là mảng con của mảng b

---

**ARRAY-SUBARR-04:** Tìm các mảng con có chứa số dương chẵn liên tiếp

---

**ARRAY-SUBARR-05:** Tìm mảng con có tổng lớn nhất, trả về con số tổng

---

**ARRAY-SUBARR-06:** Tìm mảng con có tổng lớn nhất, trả về mảng con đầu tiên

---

**ARRAY-SUBARR-07:** Tìm mảng con có số âm dương xen kẽ dài nhất

## 11. Bài tập tổng hợp luyện tư duy

---

**LOGIC-01:** Tìm 2 số có tổng bằng số cho trước

---

**LOGIC-02:** Tìm số trong mảng cách xa với số cho trước nhất

---

**LOGIC-03:** Tìm số bị thiếu cho trong một mảng từ [5..n]

---

**LOGIC-04:** Thống kê số lần xuất hiện của các số trong mảng

---

**LOGIC-05:** Tìm số có lần xuất hiện nhiều nhất

---

**LOGIC-06:** Tìm ký tự xuất hiện nhiều nhất trong câu (trừ khoảng trắng)

---

---

### Khoá học Javascript cho người mới bắt đầu 2021 🎉

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>