# Mô hình 3 khối của 10 BT JS3

## Bài 1:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bởi dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`**và**`map()`**.

Duyệt qua từng phần tử của mảng và cộng giá trị của các số dương lại với nhau.

Lưu giá trị tổng vào một biến.

**Khối 3:** Đầu ra

Hiển thị giá trị tổng của các số dương lên màn hình.

## Bài 2:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức `**split()`**và **`map()`**.

Duyệt qua từng phần tử của mảng và tăng biến đếm **`count`** nếu giá trị là số dương.

**Khối 3:** Đầu ra

Hiển thị giá trị biến đếm **`count`** lên màn hình.

## Bài 3:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và `**map()`**.

Tìm số nhỏ nhất trong mảng bằng cách sử dụng phương thức **`Math.min()`**.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 4:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Sử dụng phương thức **`filter()`** để lọc ra các giá trị số dương trong mảng.

Tìm số dương nhỏ nhất trong mảng bằng cách sử dụng phương thức **`Math.min()`**.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 5:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Tìm giá trị chẵn cuối cùng trong mảng bằng cách sử dụng vòng lặp, phương thức **`reverse()`** và **`find()`**.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 6:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Yêu cầu người dùng nhập vào hai vị trí muốn đổi chỗ giá trị.

Hoán đổi giá trị hai vị trí vừa nhập vào trong mảng.

Hiển thị mảng sau khi đã hoán đổi trên màn hình.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 7:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Sắp xếp mảng theo thứ tự tăng dần sử dụng phương thức **`sort()`**.

Hiển thị mảng đã sắp xếp trên màn hình.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 8:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng bằng cách tách chuỗi nhập vào thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Duyệt qua các phần tử trong mảng và kiểm tra xem chúng có phải là số nguyên tố hay không. Nếu có, trả về số nguyên tố đầu tiên trong mảng. Nếu không, trả về -1.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 9:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

Yêu cầu người dùng nhập vào một chuỗi chứa các số thực, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng số nguyên bằng cách tách chuỗi chứa các số nguyên thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Tạo một mảng số thực bằng cách tách chuỗi chứa các số thực thành các giá trị số thực, sử dụng phương thức **`split()`** và **`map()`**.

Duyệt qua các phần tử trong mảng số thực và kiểm tra xem chúng có phải là số nguyên hay không. Nếu có, tăng biến đếm số nguyên lên 1.

Hiển thị kết quả trên màn hình.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.

## Bài 10:

**Khối 1:** Nhập dữ liệu

Yêu cầu người dùng nhập vào một chuỗi chứa các số nguyên, các số cách nhau bằng dấu phẩy.

Lưu chuỗi vừa nhập vào biến.

**Khối 2:** Xử lý dữ liệu

Tạo một mảng số nguyên bằng cách tách chuỗi chứa các số nguyên thành các giá trị số nguyên, sử dụng phương thức **`split()`** và **`map()`**.

Duyệt qua các phần tử trong mảng và kiểm tra xem chúng có phải số dương hay số âm.

Tăng biến đếm số dương hay số âm tương ứng lên 1.

So sánh số lượng số dương và số lượng số âm để xác định số nào nhiều hơn.

Hiển thị kết quả trên màn hình.

**Khối 3:** Đầu ra

Hiển thị kết quả tính toán lên màn hình.