**Video Cần học trước: Video 1 - 5 của Java Level 2**

**Video 1 - 5 của Git**

**[Cơ bản 1] Tìm phần tử lớn nhất trong mảng**

**1. Mục tiêu:**

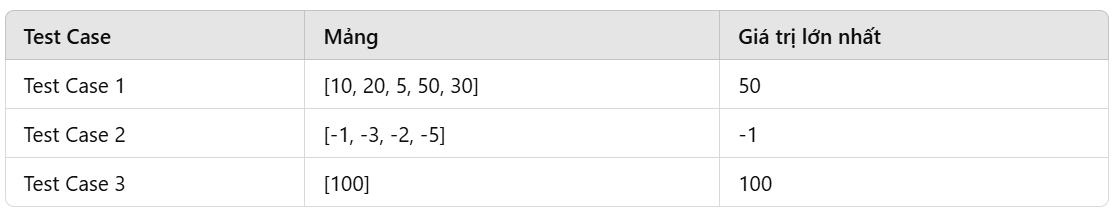
* Giúp sinh viên hiểu cách làm việc với mảng một chiều trong Java
* Luyện tập sử dụng vòng lặp for để duyệt qua các phần tử của mảng
* Làm quen với việc so sánh và tìm kiếm giá trị trong tập hợp dữ liệu

**2. Mô tả:**

Sinh viên sẽ viết chương trình cho phép người dùng nhập một mảng số nguyên. Chương trình sẽ duyệt qua các phần tử của mảng để tìm phần tử có giá trị lớn nhất, sau đó in kết quả ra màn hình.

***Yêu cầu:***

* Khởi tạo mảng:
  + Người dùng khai báo mảng và gán giá các giá trị trực tiếp cho mảng đó
* Xử lý:
  + Sử dụng một biến để lưu trữ giá trị lớn nhất, khởi tạo bằng giá trị của phần tử đầu tiên trong mảng
  + Duyệt qua từng phần tử trong mảng bằng vòng lặp for, so sánh để cập nhật giá trị lớn nhất khi cần
* Đầu ra:
  + Hiển thị giá trị lớn nhất trong mảng
* Các test case



max

max

max

max

max = 18

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | 1 | 18 | 9 |

max = 12

max = 18

max = 18

max = 12

**[Cơ bản 2] Tính tổng các phần tử trong mảng**

**1. Mục tiêu:**

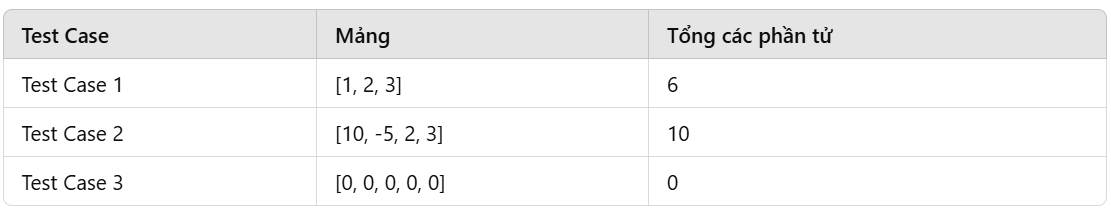
* Giúp sinh viên hiểu cách duyệt qua các phần tử của mảng trong Java
* Rèn luyện kỹ năng sử dụng vòng lặp để tính toán trên các phần tử mảng
* Làm quen với việc khởi tạo và cập nhật giá trị của biến khi xử lý dữ liệu trong mảng

**2. Mô tả:**

Sinh viên sẽ viết một chương trình nhập vào một mảng số nguyên từ bàn phím, sau đó tính tổng các phần tử trong mảng. Kết quả tổng sẽ được in ra màn hình.

***Yêu cầu:***

* Nhập dữ liệu mảng:
  + Nhập kích thước của mảng từ người dùng
  + Nhập giá trị cho từng phần tử của mảng từ bàn phím
* Xử lý:
  + Sử dụng một biến sum để lưu trữ tổng, khởi tạo giá trị ban đầu là 0
  + Duyệt qua từng phần tử trong mảng bằng vòng lặp và cộng dồn giá trị của mỗi phần tử vào biến sum
* Đầu ra:
  + Hiển thị tổng các phần tử của mảng sau khi hoàn tất tính toán
* Các test case



**[Cơ bản 3] Đếm số phần tử chẵn trong mảng**

**1. Mục tiêu:**

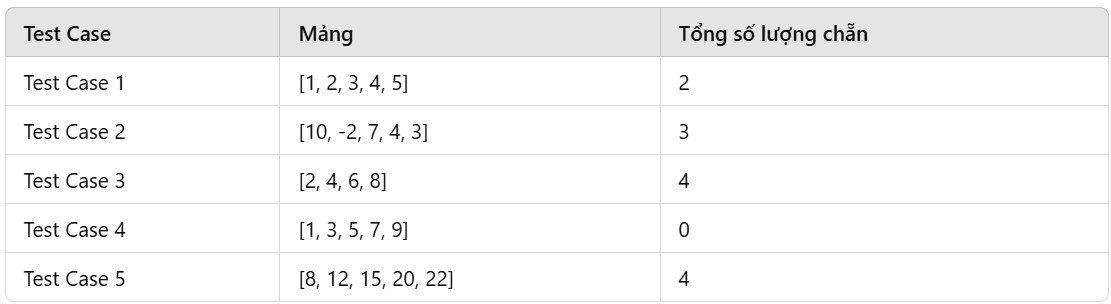
* Giúp sinh viên hiểu cách duyệt qua các phần tử của mảng trong Java
* Rèn luyện kỹ năng sử dụng vòng lặp để tính toán trên các phần tử mảng
* Làm quen với việc khởi tạo và cập nhật giá trị của biến khi xử lý dữ liệu trong mảng

**2. Mô tả:**

Sinh viên sẽ viết một chương trình nhập vào một mảng số nguyên từ bàn phím, sau đó tính tổng **số lượng** các phần tử là số chẵn trong mảng. Kết quả tổng **số lượng** sẽ được in ra màn hình.

***Yêu cầu:***

* Nhập dữ liệu mảng:
  + Nhập kích thước của mảng từ người dùng
  + Nhập giá trị cho từng phần tử của mảng từ bàn phím
* Xử lý:
  + Sử dụng một biến countEven để lưu trữ số lượng phần tử chẵn, khởi tạo giá trị ban đầu là 0
  + Duyệt qua từng phần tử trong mảng bằng vòng lặp. Nếu đó là số chẵn thì tăng countEven lên 1 đơn vị
* Đầu ra:
  + Hiển thị tổng số lượng các phần tử chẵn của mảng sau khi hoàn tất tính toán
* Các test case:



**[Cơ bản 4] Đảo ngược mảng**

**1. Mục tiêu:**

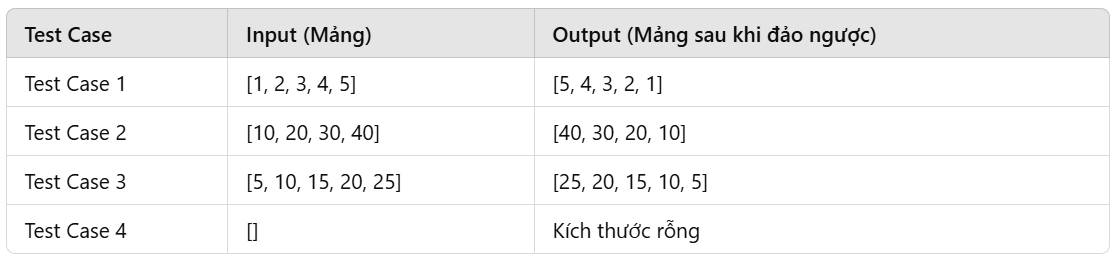
* Giúp sinh viên hiểu cách làm việc với chỉ số (index) của mảng trong Java
* Rèn luyện kỹ năng sử dụng vòng lặp để thao tác với các phần tử của mảng
* Làm quen với việc hoán đổi giá trị giữa các phần tử trong mảng mà không cần mảng phụ

**2. Mô tả:**

Sinh viên sẽ viết một chương trình nhập vào một mảng số nguyên từ bàn phím. Chương trình thực hiện đảo ngược thứ tự của các phần tử trong mảng và hiển thị kết quả sau khi đảo ngược.

***Yêu cầu:***

* Nhập dữ liệu mảng:
  + Nhập kích thước của mảng từ người dùng
  + Nhập giá trị cho từng phần tử của mảng từ bàn phím
* Xử lý:
  + Nếu mảng có kích thước rỗng thì thông báo “**Kích thước rỗng**”
  + Thực hiện đảo ngược mảng bằng cách sử dụng vòng lặp và hoán đổi các phần tử ở đầu và cuối mảng (tức là phần tử arr[i] với arr[n-i-1], trong đó n là độ dài mảng)
  + Chỉ sử dụng một mảng duy nhất, không dùng mảng phụ
* Đầu ra:
  + Hiển thị mảng ban đầu và mảng sau khi đảo ngược
* Các test case



**[Khá 5] Tìm phần tử lẻ lớn nhất và nhỏ nhất**

**1. Mục tiêu:**

* Giúp sinh viên hiểu cách duyệt mảng và tìm kiếm phần tử theo điều kiện cụ thể
* Làm quen với việc xử lý các trường hợp đặc biệt (ví dụ: không có phần tử thỏa mãn điều kiện)
* Tăng cường kỹ năng viết chương trình với các cấu trúc điều kiện và vòng lặp trong Java

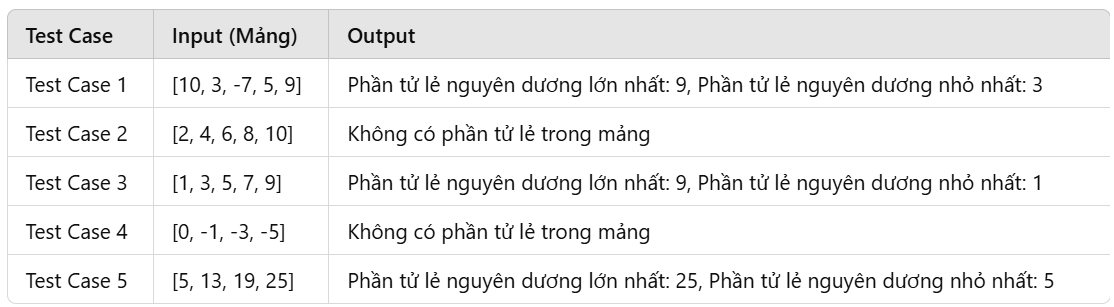
**2. Mô tả:**

Viết chương trình trong Java để tìm phần tử **lẻ nguyên dương lớn nhất** và **lẻ nguyên dương nhỏ nhất** trong một mảng số nguyên.

Nếu mảng không chứa phần tử lẻ, thông báo rằng không có phần tử lẻ trong mảng

***Yêu cầu:***

* Nhập liệu
  + Người dùng nhập kích thước mảng và các giá trị phần tử của mảng
* Chương trình phải tìm và xuất ra:
  + Giá trị của phần tử **lẻ nguyên dương lớn nhất**
  + Giá trị của phần tử **lẻ nguyên dương nhỏ nhất**
* Nếu mảng không có số nguyên dương lẻ, in thông báo: **"Không có phần tử lẻ trong mảng”**
* Các test case



**[Khá 6] Tìm phần tử xuất hiện nhiều nhất trong mảng**

**1. Mục tiêu:**

* Giúp sinh viên hiểu cách duyệt và thao tác trên mảng trong Java
* Nâng cao kỹ năng viết chương trình giải bài toán có tính ứng dụng thực tế

**2. Mô tả:**

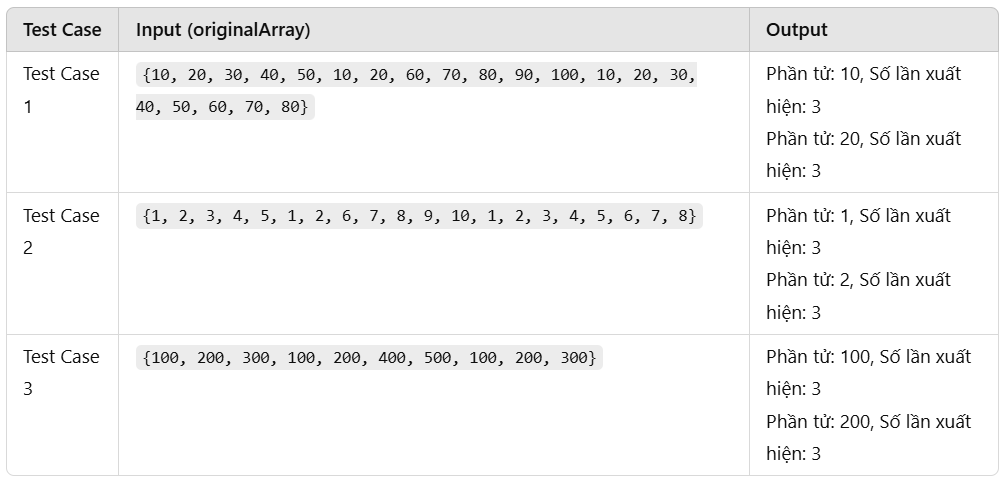
Viết chương trình trong Java để tìm phần tử **xuất hiện nhiều nhất** trong một mảng gồm các số **nguyên dương**.

Nếu có nhiều phần tử có cùng số lần xuất hiện nhiều nhất, in ra tất cả các phần tử đó.

Sử dụng phương pháp mảng đánh dấu(Marking Array) để giải quyết bài toán cho số không âm có giá trị nhỏ.

***Yêu cầu:***

* Khởi tạo mảng
  + Khởi tạo mảng **markingArray[1000]**
  + Khởi tạo mảng **originalArray[20]** và gán giá trị các phần tử trong mảng trong khoảng từ 1 đến 1000.
  + Học viên khởi tạo trực tiếp 20 giá trị trong khoảng từ 1-1000 mà không cần nhập từ bàn phím
* Chương trình phải thực hiện:
  + Duyệt qua phần tử của mảng **originalArray** và dùng phương pháp Marking Array để đếm và lưu số lần xuất hiện vào mảng **markingArray**
  + Tìm phần tử hoặc các phần tử có số lần xuất hiện nhiều nhất.
* Xuất ra phần tử xuất hiện nhiều nhất cùng với số lần xuất hiện.
* Các test case



**[Giỏi 7] Sắp xếp mảng cho các số chẵn đứng trước, số lẻ đứng sau (Giữ nguyên thứ tự trong từng nhóm)**

**1. Mục tiêu:**

* Nâng cao kỹ năng xử lý dữ liệu mảng trong Java
* Rèn luyện khả năng sắp xếp dữ liệu theo điều kiện cụ thể mà không phá vỡ thứ tự ban đầu của từng nhóm phần tử

**2. Mô tả:**

Viết chương trình trong Java để sắp xếp một mảng số nguyên sao cho:

* Các phần tử **chẵn** nằm ở đầu mảng, các phần tử **lẻ** nằm ở cuối mảng
* Thứ tự ban đầu của các phần tử trong nhóm **chẵn** và nhóm **lẻ** phải được giữ nguyên

***Yêu cầu:***

* Người dùng nhập số phần tử của mảng và giá trị của các phần tử
* Chương trình thực hiện sắp xếp mảng sao cho các số **chẵn** đứng trước, các số **lẻ** đứng sau, **giữ nguyên thứ tự trong từng nhóm**
* Kết quả phải được in ra mảng sau khi sắp xếp
* Nếu mảng không hợp lệ (có kích thước bằng 0), chương trình phải hiển thị thông báo lỗi “**Mảng không có phần tử**”
* Sau đây là một số test case:

