**Bài Tập**

**1: Tham số, đối số và giá trị trả về.**

**Bài tập 1: Tính tổng của hai số nguyên**

**Mô tả:** Viết một phương thức có tên sum nhận hai tham số kiểu int và trả về tổng của chúng.

**Yêu cầu:**

* Tạo một lớp Main.
* Viết phương thức sum(int a, int b) trả về tổng của a và b.
* Gọi phương thức sum từ phương thức main và in ra kết quả.

**Bài tập 2: Tính diện tích hình chữ nhật**

**Mô tả:** Viết một phương thức có tên calculateRectangleArea nhận hai tham số kiểu double là chiều dài và chiều rộng, và trả về diện tích của hình chữ nhật.

**Yêu cầu:**

* Tạo một lớp Main.
* Viết phương thức calculateRectangleArea(double length, double width) trả về diện tích của hình chữ nhật.
* Gọi phương thức từ phương thức main và in ra kết quả.

**Bài tập 3: Tính giá trị trung bình của ba số thực**

**Mô tả:** Viết một phương thức có tên calculateAverage nhận ba tham số kiểu double và trả về giá trị trung bình của chúng.

**Yêu cầu:**

* Tạo một lớp Main.
* Viết phương thức calculateAverage(double a, double b, double c) trả về giá trị trung bình của a, b và c.
* Gọi phương thức từ phương thức main và in ra kết quả.

**Bài tập 4: Kiểm tra số nguyên tố**

**Mô tả:** Viết một phương thức có tên isPrime nhận một tham số kiểu int và trả về true nếu số đó là số nguyên tố, ngược lại trả về false.

**Yêu cầu:**

* Tạo một lớp Main.
* Viết phương thức isPrime(int number) sử dụng câu lệnh if-else để kiểm tra số nguyên tố.
* Gọi phương thức từ phương thức main và in ra kết quả.

**2. Phạm vi biến.**

* **Phạm vi biến cục bộ (Local Variable Scope)**

#### **Bài tập 1: Tính tổng của hai số**

* **Mô tả:** Viết một phương thức để tính tổng của hai số nguyên, sử dụng biến cục bộ.

#### **Bài tập 2: Kiểm tra số chẵn lẻ**

* **Mô tả:** Viết một phương thức để kiểm tra xem một số có phải là số chẵn hay không, sử dụng biến cục bộ.
* **Phạm vi biến thể hiện (Instance Variable Scope)**

#### **Bài tập 3: Quản lý thông tin sinh viên**

* **Mô tả:** Tạo một lớp Student có các biến thể hiện để lưu trữ thông tin về tên và tuổi của sinh viên. Viết phương thức để hiển thị thông tin của sinh viên.

#### **Bài tập 4: Quản lý tài khoản ngân hàng**

* **Mô tả:** Tạo một lớp BankAccount có các biến thể hiện để lưu trữ thông tin về số tài khoản và số dư. Viết phương thức để nạp tiền và hiển thị số dư.
* **Phạm vi biến lớp (Class Variable Scope)**

#### **Bài tập 5: Đếm số đối tượng được tạo ra**

* **Mô tả:** Tạo một lớp Counter với biến lớp để đếm số lượng đối tượng được tạo ra. Hiển thị số lượng đối tượng mỗi khi một đối tượng mới được tạo.

#### **Bài tập 6: Quản lý số lượng sản phẩm**

* **Mô tả:** Tạo một lớp Product với biến lớp để lưu trữ số lượng sản phẩm hiện có. Viết phương thức để thêm sản phẩm và hiển thị tổng số sản phẩm.
* **Phạm vi biến tham số (Parameter Variable Scope)**

#### **Bài tập 7: Tính lũy thừa**

* **Mô tả:** Viết một phương thức nhận hai tham số kiểu int và trả về giá trị lũy thừa của số thứ nhất với số thứ hai.

#### **Bài tập 8: So sánh hai số**

* **Mô tả:** Viết một phương thức nhận hai tham số kiểu int và trả về số lớn hơn.

**3. Cơ bản về chia để trị và đệ quy.**

**Bài tập Chia để trị (Divide and Conquer):**

1. **Tìm kiếm phần tử lớn nhất trong mảng:**
   * Viết một phương thức sử dụng chia để trị để tìm phần tử lớn nhất trong mảng nguyên dương.
   * Gợi ý: Chia mảng thành hai nửa, tìm phần tử lớn nhất của mỗi nửa, sau đó so sánh để lấy phần tử lớn nhất của toàn mảng.
2. **Tính tổng các số nguyên tố trong mảng:**
   * Viết một phương thức sử dụng chia để trị để tính tổng các số nguyên tố trong một mảng số nguyên.
   * Gợi ý: Chia mảng thành hai nửa, tính tổng số nguyên tố của mỗi nửa và cộng tổng này lại.
3. **Đếm số lần xuất hiện của một phần tử trong mảng:**
   * Viết một phương thức sử dụng chia để trị để đếm số lần xuất hiện của một phần tử trong một mảng số nguyên.
   * Gợi ý: Chia mảng thành hai nửa, đệ quy đếm số lần xuất hiện của phần tử trong mỗi nửa và cộng các kết quả lại.
4. **Tìm số lớn thứ k trong mảng:**
   * Viết một phương thức sử dụng chia để trị để tìm số lớn thứ k trong một mảng số nguyên.
   * Gợi ý: Chia mảng thành hai nửa, tìm số lớn thứ k của mỗi nửa và so sánh để chọn số lớn thứ k của toàn mảng.
5. **Tìm số lớn nhất và nhỏ nhất trong mảng:**
   * Viết một phương thức sử dụng chia để trị để tìm số lớn nhất và nhỏ nhất trong một mảng số nguyên.
   * Gợi ý: Chia mảng thành hai nửa, đệ quy tìm số lớn nhất và nhỏ nhất của mỗi nửa, sau đó so sánh để có kết quả cuối cùng.

**Bài tập Đệ quy (Recursion):**

1. **Tính tổng các chữ số của một số nguyên dương:**
   * Viết một phương thức đệ quy để tính tổng các chữ số của một số nguyên dương.
   * Gợi ý: Dùng đệ quy để lấy từng chữ số của số nguyên và cộng chúng lại.
2. **Tìm ước chung lớn nhất (UCLN) của hai số nguyên:**
   * Viết một phương thức đệ quy để tính ước chung lớn nhất của hai số nguyên.
   * Gợi ý: Sử dụng thuật toán Euclid để tìm UCLN của hai số.
3. **Tìm số Fibonacci thứ n:**
   * Viết một phương thức đệ quy để tính số Fibonacci thứ n.
   * Gợi ý: Dùng đệ quy để tính số Fibonacci bằng cách gọi lại chính mình với các số Fibonacci trước đó.
4. **Kiểm tra tính đối xứng của một chuỗi:**
   * Viết một phương thức đệ quy để kiểm tra xem một chuỗi có phải là chuỗi đối xứng hay không.
   * Gợi ý: So sánh ký tự đầu tiên và cuối cùng của chuỗi, sau đó gọi đệ quy với phần còn lại của chuỗi.
5. **Đảo ngược một chuỗi:**
   * Viết một phương thức đệ quy để đảo ngược một chuỗi.
   * Gợi ý: Sử dụng đệ quy để đảo ngược phần còn lại của chuỗi sau khi đảo ngược ký tự đầu tiên.

**4. Thuật toán sắp xếp trong java**

1. **Sắp xếp mảng số nguyên bằng thuật toán Bubble Sort:**

* Yêu cầu: Viết chương trình sắp xếp một mảng các số nguyên bằng thuật toán Bubble Sort.
* Hướng dẫn: Đọc dữ liệu từ người dùng hoặc sử dụng mảng đã cung cấp. Sau đó, sử dụng thuật toán Bubble Sort để sắp xếp mảng và in ra mảng đã sắp xếp.

1. **Sắp xếp danh sách chuỗi bằng thuật toán Selection Sort:**

* Yêu cầu: Viết chương trình sắp xếp một danh sách các chuỗi bằng thuật toán Selection Sort.
* Hướng dẫn: Tạo một danh sách chuỗi và sử dụng thuật toán Selection Sort để sắp xếp danh sách này theo thứ tự từ điển (tăng dần). In ra danh sách đã sắp xếp.

1. **Sắp xếp mảng số thực bằng thuật toán Insertion Sort:**

* Yêu cầu: Viết chương trình sắp xếp một mảng các số thực bằng thuật toán Insertion Sort.
* Hướng dẫn: Tạo một mảng các số thực và sử dụng thuật toán Insertion Sort để sắp xếp mảng này. Sau đó, in ra mảng đã sắp xếp.

1. **Sắp xếp mảng số nguyên bằng thuật toán Merge Sort:**

* Yêu cầu: Viết chương trình sắp xếp một mảng các số nguyên bằng thuật toán Merge Sort.
* Hướng dẫn: Đọc dữ liệu từ người dùng hoặc sử dụng mảng đã cung cấp. Sử dụng thuật toán Merge Sort để sắp xếp mảng và in ra mảng đã sắp xếp.

1. **Sắp xếp mảng số nguyên bằng thuật toán Quick Sort:**

* Yêu cầu: Viết chương trình sắp xếp một mảng các số nguyên bằng thuật toán Quick Sort.
* Hướng dẫn: Tạo một mảng số nguyên và sử dụng thuật toán Quick Sort để sắp xếp mảng này. In ra mảng đã sắp xếp sau khi hoàn thành thuật toán.

**5.Cơ bản về thuật toán và độ phức tạp của thuật toán.**

**Bài 1:**

Viết một chương trình cho phép xác định ký tự xuất hiện nhiều nhất trong một chuỗi.

Chương trình cho phép nhập vào một chuỗi và hiển thị ký tự có số lần xuất hiện nhiều nhất trong chuỗi đó.

Tính độ phức tạp của thuật toán vừa triển khai.