

Thực hành các cấu trúc lệnh cơ bản: vòng lặp, rẽ nhánh

1. Bài toán dạng 1: Kiểm tra_Kết luận;

1.1. Phương pháp giải:

(1) Đưa ra giả thiết;

(2) Kiểm tra giả thiết - nếu thấy sai: $///\Rightarrow$ kết luận và dừng việc kiểm tra tiếp

1.2. Ví dụ:

Nhập vào số **nguyên n** \Rightarrow ngto **hay ko phải là nguyên tố**

(1) boolean ktra=true; // giả thiết n là số nt

(2)

```
for (int i=2; i<=Math.sqrt(n);i++) {
```

```
    if (n%i==0) {
```

```
        ktra=false; //
```

```
        break;
```

```
    }
```

```
} // Khi nào thì bỏ {} được?
```

```
if (ktra)
```

```
    System.out.println(n + " là số nt");
```

```
else
```

```
    System.out.println(n + " ko là số nt");
```

1.3. Bài tập áp dụng (cần làm với nhiều loại vòng lặp khác nhau):

(1) Mảng int [] A={5, 7, 9, 10,16} tạo thành dãy **tăng dần** hay ko? \Rightarrow yes/No

(2) Mảng int [] A={5, -1, 9, -10,16} tạo thành dãy **đan dấu** hay ko? \Rightarrow yes/No

(3) Mảng `int [] A={ 5, 1, 9, 10, 16}` ///ktra xem có phải tất cả các phần tử của A đều là số nguyên tố hay ko?

2. Dạng 2: Kiểm tra_Cập nhật giả thiết

2.1. Phương pháp giải

- (1) Giả thiết;
- (2) Kiểm tra giả thiết nếu giả thiết sai: **cập nhật lại** giả thiết và tiếp tục,...

2.2. Ví dụ:

tìm giá trị **LN** trong một mảng

```
int [] A={ 12, 3, 34, 43,434,38,55 }; // In ra giá trị LN (NN) //
```

```
(1) int max=A[0]; // Giả thiết A[0] là lớn nhất; // max=0;
```

```
(2)
```

```
    for (int i=1; i<A.length; i++) {
```

```
        if (A[i]>max)
```

```
            max=A[i]; // Cập nhật giả thiết
```

```
    }
```

```
    System.out.println("Max =" + max);
```

2.3. Bài tập áp dụng

(1) Cho mảng `int [] A={5, 1, 9, 10, 16, 4, 6, 8, 3}`. // Hãy xác định và in dãy con tăng dần dài nhất //

(2) Cho mảng `int [] A={5, 1, 9, 10,16, 4,6, 8, 3}`. // Hãy xác định và in dãy con tăng dần dài nhất có tổng các phần tử lớn hơn 20 //

(3) Cho mảng `int [] A={5, 1, 9, 10,16, 4,6, 8, 3}`. // Hãy xác định và in dãy con dài có các số chẵn liên tiếp dài nhất

3. Dạng 3: Mở rộng của Dạng 2

3.1. Phương pháp giải

(1) Giả thiết; Khởi tạo biến trạng thái

(2) Kiểm tra giả thiết:

Nếu giả thiết đúng thì cập nhật biến trạng thái

Nếu giả thiết sai thì cập nhật lại giả thiết và **Reset** lại biến trạng thái.

3.2. Ví dụ:

Hãy xác định giá trị **LN** và in ra số lần xuất hiện cùng các ví trị xuất hiện của nó trong mảng:

```
int [] A={ 12, 3, 434, 12, 43, 434, 38,55 };
```

```
int [] pos = new int[A.length];
```

```
(1) int max=A[0]; // Giả thiết A[0] LN
```

```
int cnt=1; // Trạng thái xuất hiện 1 lần
```

```
pos[cnt-1]=0;
```

```
for (int i=1; i<A.length;i++) {
```

```
    if (max==A[i]) { // Cập nhập biến trạng thái
```

```
        cnt++; pos[cnt-1]=i;
```

```
    }
```

```
    else if (max<A[i]) {
```

```
        max=A[i]; // Cập nhập giả thiết
```

```
        cnt=1; pos[cnt-1]=i; // Reset lại biến trạng thái
```

```
    }
```

```
}
```

```

System.out.println("Max = "+ max);
System.out.println("Số lần xuất hiện: "+ cnt);
System.out.println("Xuất hiện ở các vị trí sau: ");
for (int i=0;i<cnt;i++) System.out.print(pos[i]+" ");
/* Thay đổi đoạn in kết quả theo yêu cầu sau:
(1) Nếu chỉ xuất hiện 1 lần: Max = ... ở vị trí: ....
(2) Nếu xuất hiện nhiều lần:
    Max = ... Số lần: ....
    Ở các vị trí: 3, 5, 6, ...10. */

```

3.3. Bài tập áp dụng:

- (1) Cho mảng `int [] A={5, 3, 7, 8, 4, 6, 8, 3}`. // Hãy xác định và in **tất cả** các dãy con tăng dần dài nhất cùng các vị trí bắt đầu xuất hiện của nó
- (2) Cho mảng `int [] A={5, 1, 9, 10,16, 4,6, 8, 3}`. // Hãy xác định và in **tất cả** các dãy con tăng dần dài nhất có tổng các phần tử lớn hơn 100 cùng các vị trí bắt đầu xuất hiện của nó
- (3) Cho mảng `int [] A={5, 1, 9, 10,16, 4,6, 8, 3}`. // Hãy xác định và in ra **tất cả các** dãy con dài nhất gồm các phần tử là các số chẵn.

4. Các thuật toán sắp xếp với các cấu trúc lặp khác nhau

// Viết đc chg trình :

// Chạy được từng bước trên giấy trên một mảng cụ thể có 8 phần tử

5. . Xây dựng **bảng băm** sử dụng mảng

5.1. Không chèn thêm khi có xung đột

5.2. Chèn đè khi xảy ra xung đột

6. Cài đặt Stack sử dùng mảng

- Ứng dụng Stack để biến đổi một số x từ hệ 10 sang hệ 2.

7. Cài đặt hàng đợi dùng mảng

- Mở rộng: hàng đợi quay vòng

8. String, StringBuffer, StringTokenizer

8.1. Cho một chuỗi ký tự gồm các từ cách nhau bởi dấu cách, chấm, phẩy, hỏi chấm,...
=> Hãy xác định và in ra tất cả các từ dài nhất (các từ không được lặp lại).

8.2. Cho đoạn văn bản gồm các câu cách nhau bởi dấu chấm. Hãy chuẩn hóa văn bản đó bằng cách viết hoa đầu câu, trước dấu chấm không có dấu cách, sau dấu chấm phải có dấu cách.

8.3. Cho một mảng chuỗi ký tự, chứa danh sách họ tên sinh viên của lớp LA2023. Hãy đếm số sinh viên có tên là Quân và họ Nguyễn.

9. Regular Expression

9.1. Kiểm tra username:

Username có độ dài từ 6 đến 12 ký tự, không có khoảng trắng và không dấu hai chấm.

`[a-z0-9_]{6,12}$`

9.2. Kiểm tra email

- Bắt đầu bằng chữ cái.

- Chỉ chứa chữ cái, chữ số và dấu gạch ngang (-).

- Chứa một ký tự @, sau @ là tên miền.

- Tên miền có thể là domain.xxx.yyy hoặc domain.xxx. Trong đó xxx và yyy là các chữ cái và có độ dài từ 2 trở lên.

`^[a-zA-Z][\w-]+@([\w]+\.[\w]+|[\w]+\.[\w]{2,})\.[\w]{2,})$`