**STRING – ARRAYS – OOP**

***NgoThuan made this :P***

**String**

**Task 1: [GreeksForGreeks]**

Viết chương trình chỉ in các từ có số lượng kí tự chẵn:

Input: s = “This is a java language”

Output: This is java language

Giải thích:

This: 4 kí tự, is: 2 kí tự, java: 4 kí tự, language: 8 kí tự -> Được in ra.

a: 1 kí tự -> không được in ra.

**Task 2: [GreeksForGreeks]**

Viết chương trình xóa các số không dư thừa ở đầu:

Input: vcb = “00001026825541=))”

Output: 1026825541=))

**Task 3: [GreeksForGreeks]**

Viết chương trình sắp xếp chuỗi theo bảng chữ cái

Input: str = “sdfadmcdpow”

Output: acdddfmopsw

**Task 4: [GreeksForGreeks]**

Viết chương trình tính tổng 2 chuỗi (String) và xuất ra chuỗi (String)

Input: a = “1739282”, b = “2312908”

Output: 4052190

**Task 5: [freetuts.net]**

Viết chương trình nhập từ bàn phím một chuỗi không quá 80 ký tự và một ký tự bất kỳ. Đếm và in ra màn hình số lần xuất hiện của ký tự đó trong chuỗi vừa nhập.

**Task 6: [freetuts.net]**

Viết chương trình nhập vào một chuỗi bất kỳ bao gồm cả số, ký tự thường và ký tự hoa từ bàn phím. Sau đó đếm và in ra số ký tự thường và ký tự hoa và số có trong chuỗi đó.

Bài tập thêm: <https://java-vku.vercel.app/advanced-exercises/>

**Arrays**

**Task 1: [freetuts.net]**

Viết chương trình thực hiện các công việc sau:

Nhập liệu cho mảng A có n phần tử nguyên từ bàn phím. Sắp xếp mảng số đã nhập tăng dần theo giá trị của các phần tử có trong mảng.

In ra màn hình mảng số ban đầu và mảng đã sắp xếp theo thứ tự tăng dần.

*Yêu cầu kỹ thuật*: Chương trình phải kiểm tra n nhập vào (n >= 2 và n <= max - 1, với max là số phần tử tối đa của mảng).

**Task 2: [freetuts.net]**

Viết chương trình thực hiện các công việc sau:

Nhập liệu cho mảng có n phần tử nguyên (n > 0) từ bàn phím.

Nhập số nguyên k từ bàn phím.

Tìm kiếm phần tử đầu tiên trong mảng có giá trị bằng k và thông báo lên màn hình vị trí của phần tử đó. Nếu không có phần tử nào của mảng có giá trị bằng k thì thông báo "Trong mảng không có phần tử nào chứa giá trị cần tìm."

*Yêu cầu kỹ thuật*: Chương trình phải kiểm tra n nhập vào: nếu n <= 0 hoặc n > số phần tử tối đa của mảng thì yêu cầu nhập lại số phần tử cho đến khi thỏa mãn điều kiện.

**Task 3: [freetuts.net]**

Viết chương trình thực hiện các công việc sau:

Nhập m, n là số dòng và 2 số cột của 2 ma trận 2 chiều A và B từ bàn phím.

Nhập giá trị cho các phần tử cho 2 ma trận này.

Tính và in ra màn hình ma trận C là tổng của 2 ma trận này.

*Yêu cầu kỹ thuật*: Kiểm tra số dòng, số cột nhập vào không được nhỏ hơn 1.

**Task 4: [freetuts.net]**

Một ma trận được gọi là ma trận thưa nếu số phần tử có giá trị bằng 0 nhiều hơn số phần tử khác 0. Viết chương trình thực hiện các công việc sau:

Nhập m, n là số dòng và số cột của ma trận hai chiều A từ bàn phím.

Nhập giá trị các phần tử của ma trận A từ bàn phím.

Kiểm tra và thông báo lên màn hình ma trận vừa nhập là ma trận thưa hay ma trận không thưa.

Yêu cầu kỹ thuật: Kiểm tra số dòng, số cột nhập vào không được nhỏ hơn 1.

**Task 5: [freetuts.net]**

Một ma trận được gọi là ma trận đối xứng trước hết nó phải là ma trận vuông (có số dòng và số cột bằng nhau) và các phần tử của nó đối xứng nhau qua đường chéo chính. Viết chương trình nhập từ bàn phím các phần tử của ma trận A, kích thước m dòng, n cột (1 <= m, n <= 5). Kiểm tra xem ma trận vừa nhập có phải là ma trận đối xứng hay không?

*Hướng dẫn*: Giả sử chúng ta có một ma trận vuông có 3 dòng, 3 cột thì chúng ta gọi ma trận này là ma trận vuông bậc 3. Hình dưới đây minh họa đường chéo phụ và đường chéo chính như sau: (Méo cho xem đâu :D)

Các phần tử nằm trên đường chéo chính có đặc điểm chỉ số dòng bằng chỉ số cột. Ví dụ trong hình trên chúng ta thấy có các phần tử nằm trên đường chéo chính như a11, a12, ..., ann, các phần tử này được gọi là các phần tử chéo.

Để kiểm tra ma trận A có phải là ma trận đối xứng hay không thì cần thực hiện các bước sau:

Ma trận A nhập vào phải luôn luôn là một ma trận vuông.

Kiểm tra phần tử nằm dưới đường chéo chính có bằng phần tử đối xứng với nó qua đường chéo chính hay không (tức là kiểm tra các phần tử A[i][j] có bằng A[j][i] không, với i, j chạy từ 0 đến n).

**OOP**

Bài tập thêm về constructor: <https://java-vku.vercel.app/oop-constructor/>

**GIỮA KỲ**

Các bài thi mẫu [VKU]: <https://java-vku.vercel.app/midterm-exam/> (Bao gồm ArrayList)