

## JAVA EXERCISES PART 2

**Bài 1.** Đọc vào dãy số  $a$  có  $n$  phần tử chỉ gồm các số nguyên dương. Kiểm tra mảng  $a$  có phải là mảng đối xứng hay không (ví dụ: 15 2 1 2 15 là mảng đối xứng).

**Bài 2.** Đọc vào một dãy số  $a$  có  $n$  phần tử là các số nguyên. Đếm số lần xuất hiện của từng phần tử trong mảng. In ra phần tử và số lần xuất hiện của phần tử có số lần xuất hiện nhiều nhất trong  $a$ .

**Bài 3.** Đọc vào một dãy số  $a$  có  $n$  phần tử là các số nguyên. Nhập vào một số  $X$ . Xác định vị trí của số nguyên tố trên  $a$  có giá trị gần với  $X$  nhất.

**Bài 4.** Tạo ngẫu nhiên mảng  $a[N]$  gồm các số nguyên sau đó thực hiện:

- Sắp xếp dãy  $a$  theo thứ tự tăng dần
- Loại bỏ bớt các phần tử trùng nhau sao cho mỗi giá trị chỉ xuất hiện trong  $a$  duy nhất một lần.

**Bài 5.** Nhập 2 mảng  $a[N]$  và  $b[M]$  và số nguyên  $p$  ( $0 \leq p < N$ ). Hãy chèn mảng  $b[]$  vào vị trí  $p$  của  $a[]$ . Ví dụ:  $a[4]: 5\ 3\ 6\ 7$ ;  $b[3]: 2\ 9\ 11$ ;  $p = 1 \rightarrow a[7]: 5\ \underline{2\ 9\ 11}\ 3\ 6\ 7$ .

**Bài 6.** Nhập số liệu cho dãy số nguyên  $a_0, a_1, \dots, a_{n-1}$  và một giá trị  $x$ . Giả sử dãy  $a$  đã được sắp xếp theo thứ tự tăng dần. Hãy chèn giá trị  $x$  vào dãy  $a$  sao cho vẫn giữ được tính sắp xếp của mảng.

**Bài 7.** Nhập dãy  $n$  số ( $n \leq 1000$ ). Xác định đường chạy dài nhất, xuất lên màn hình vị trí phần tử đầu tiên và độ dài của đường chạy đó. Đường chạy là một dãy liên tiếp các phần tử không giảm của dãy ban đầu.

Ví dụ : Nhập dãy 1 4 2 3 1 2 6 8 3 5 7

Đường chạy dài nhất là : 4 4

**Bài 8.** Viết chương trình nhập số liệu cho ma trận các số nguyên  $A$  cấp  $m \times n$  trong đó  $m, n$  là các số tự nhiên. Sau đó tìm ma trận chuyển vị  $B = (b_{ij})$  cấp  $n \times m$ ,  $b_{ij} = a_{ji}$   $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ . Hiển thị 2 ma trận trên màn hình để tiện so sánh.

**Bài 9.** Viết chương trình tính tích 2 ma trận các số nguyên  $A[m][n]$  và  $B[n][k]$ .

**Bài 10.** Nhập vào ma trận  $A[n][m]$  có các phần tử là các số nguyên. Hãy liệt kê trên màn hình tất cả các phần tử của ma trận theo thứ tự tăng dần từ trái qua phải, từ trên xuống dưới.

**Bài 11.** Viết chương trình in ra ma trận xoắn ốc  $Q[m][n]$ . Trong đó  $m, n > 0$  tùy ý.

**Bài 12.** Viết chương trình chuyển đổi một số tự nhiên ở hệ cơ số 10 thành số ở hệ cơ số  $b$  bất kì ( $2 < b \leq 36$ ).

**Bài 13.** Sử dụng xâu ký tự để viết phương thức kiểm tra số thuận nghịch. Áp dụng cho việc liệt kê các số thuận nghịch có 9 chữ số.

**Bài 14.** Xây dựng tập thao tác với các số nguyên lớn (nhiều hơn 20 chữ số)

- a) Tổng hai số nguyên lớn
- b) Hiệu hai số nguyên lớn
- c) Tích hai số nguyên lớn
- d) Thương hai số nguyên lớn

**Bài 15.** Viết chương trình thực hiện chuẩn hoá một xâu ký tự nhập từ bàn phím hoặc từ file (loại bỏ các dấu cách thừa, chuyển ký tự đầu mỗi từ thành chữ hoa, các ký tự khác thành chữ thường).

**Bài 16.** Viết chương trình thực hiện đọc vào một xâu ký tự gồm nhiều từ và tìm từ dài nhất trong xâu đó. Từ đó xuất hiện ở vị trí nào? (Nếu có nhiều từ có độ dài giống nhau thì chọn từ đầu tiên tìm thấy).

**Bài 17.** Đọc vào một xâu họ tên theo cấu trúc: *họ...đệm...tên*; sau đó thực hiện chuyển xâu đó sang biểu diễn theo cấu trúc *tên...họ...đệm*.

Ví dụ: Triệu Văn Thân -> Thân Triệu Văn.

Áp dụng với file đầu vào có số dòng dữ liệu chưa biết trước, trong đó mỗi dòng chứa duy nhất một thông tin đầy đủ họ tên. Ghi kết quả ra file KETQUA.OUT

**Bài 18.** Tính giá trị của một số viết dưới dạng LA MÃ.

Ví dụ: MDCLXVI = 1666. M: 1000 ; D: 500 ; C: 100; L: 50; X :10 ; V: 5 ; I: 1

**Bài 19.** Viết chương trình tách các từ trong câu và in các từ theo thứ tự Alphabet. Áp dụng tách các từ cho trước trong file input.txt và ghi kết quả ra file output.txt

**Bài 20.** Tìm hợp, giao, hiệu của tập các từ xuất hiện trong hai xâu str1 và str2. Ghi kết quả lần lượt vào các file tương ứng có đuôi .OUT. Ví dụ: HOP.OUT, HIEU.OUT