

## TIN HỌC CƠ SỞ 2 - CONTEST 4

### BÀI A : SỐ CHẴN

Viết chương trình in ra các số chẵn trong một dãy số cho trước.

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi test gồm 2 dòng, dòng đầu ghi số N là số phần tử của dãy. Dòng sau ghi N số của dãy. N không quá 100, các số trong dãy đều nguyên dương và không quá 1000.

**Kết quả:** Với mỗi bộ test ghi trên một dòng lần lượt là các số chẵn của dãy số ban đầu, theo thứ tự xuất hiện.

### BÀI B : SỐ NGUYÊN TỐ TRONG DÃY

Viết chương trình in ra các số nguyên tố trong một dãy số cho trước.

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi test gồm 2 dòng, dòng đầu ghi số N là số phần tử của dãy. Dòng sau ghi N số của dãy. N không quá 100, các số trong dãy đều nguyên dương và không quá 1000.

**Kết quả:** Với mỗi bộ test ghi trên một dòng lần lượt là các số nguyên tố của dãy số ban đầu, theo thứ tự xuất hiện.

### BÀI C: MẢNG ĐỐI XỨNG

Nhập một dãy số nguyên có n phần tử (n không quá 100, các phần tử trong dãy không quá  $10^9$ ). Hãy viết chương trình kiểm tra xem dãy có phải đối xứng hay không. Nếu đúng in ra YES, nếu sai in ra NO.

**Dữ liệu vào:** Dòng đầu ghi số bộ test, mỗi bộ test gồm hai dòng. Dòng đầu là số phần tử của dãy, dòng sau ghi ra dãy đó, mỗi số cách nhau một khoảng trống.

**Kết quả:** In ra kết quả kiểm tra.

Input	Output
2	YES
4	NO
1 4 4 1	
5	
1 5 5 5 3	

### BÀI D: TAM GIÁC PASCAL

Nhập chiều cao n (không quá 20) và xây dựng tam giác vuông chứa các số  $C_n^k$  theo công thức truy hồi sau:

$$C_n^0 = C_n^n = 1$$

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

*Dòng thứ n của tam giác trên (bắt đầu từ n=0) chứa n+1 phần tử là các hệ số của nhị thức  $(a+b)^n$  và được gọi là tam giác Pascal.*

**Ví dụ:**

Input	Output
4	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

### **BÀI E: SỐ FIBONACCI**

Dãy số Fibonacci được định nghĩa theo công thức như sau:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2} \text{ với } n > 2$$

Viết chương trình tính số Fibonacci thứ n (với n không quá 92)

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi bộ test là một số nguyên n.

**Kết quả:** Với mỗi bộ test, ghi ra số Fibonacci thứ n trên một dòng.

**Ví dụ**

Input	Output
3	1
2	5
5	6765
20	

### **BÀI F: ĐOẠN TĂNG DÀI NHẤT**

Một đoạn tăng trong một dãy số nguyên là một đoạn liên tiếp trong dãy sao cho phần tử phía sau lớn hơn phần tử phía trước. Cho dãy số với n phần tử (n không quá 100, các phần tử đều không quá 1000). Viết chương trình tìm các đoạn tăng liên tiếp trong dãy mà số phần tử là nhiều nhất.

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi test gồm 2 dòng, dòng đầu ghi số N là số phần tử của dãy. Dòng sau ghi N số của dãy. N không quá 100, các số trong dãy đều nguyên dương và không quá 1000.

**Kết quả:** Với mỗi bộ test, ghi ra thứ tự bộ test. Sau đó là 1 dòng ghi độ dài của đoạn tăng dài nhất. Tiếp theo là một số dòng ghi lần lượt các đoạn tăng dài nhất, từ trái qua phải trong dãy ban đầu.

**Ví dụ:**

Input	Output
2	Test 1:
16	4
2 3 5 7 4 5 8 9 7 11 8 9 6 7 10 12	2 3 5 7
12	4 5 8 9
2 3 2 3 2 3 2 2 2 3 4 1	6 7 10 12
	Test 2:
	3
	2 3 4

### BÀI G: CHÈN MẢNG

Nhập 2 mảng (a, N) và (b, M) và số nguyên p ( $0 \leq p < M \leq N < 100$ ). Hãy chèn mảng b vào vị trí p của mảng a. **Dữ liệu vào:** Dòng đầu ghi số bộ test, mỗi bộ test gồm 3 dòng: dòng đầu ghi 3 số N,M,p. Dòng thứ 2 ghi N số của mảng a. Dòng thứ 3 ghi M số của mảng b. **Kết quả** ghi ra thứ tự bộ test và dãy số sau khi chèn.

Input	Output
1 4 3 1 5 3 6 7 2 9 11	Test 1: 5 2 9 11 3 6 7

### BÀI H: ĐẾM SỐ PHẦN TỬ LỚN HƠN SỐ ĐỨNG TRƯỚC TRONG DÃY

Cho một dãy số nguyên dương có n phần tử ( $2 \leq n \leq 50$ ). Hãy liệt kê số các phần tử trong dãy không nhỏ hơn các số đứng trước nó (tính cả phần tử đầu tiên). **Dữ liệu vào:** Dòng 1 ghi số bộ test. Mỗi bộ test ghi ra 2 dòng: dòng 1 là số phần tử, dòng thứ 2 ghi dãy số. **Kết quả** ghi ra số phần tử thỏa mãn.

Input	Output
1 7 3 5 6 8 4 2 9	5

### BÀI I: SỐ LỚN NHẤT

Viết chương trình tìm số lớn nhất trong một dãy số cho trước.

**Dữ liệu vào:** Dòng đầu ghi số bộ test. Mỗi test gồm 2 dòng, dòng đầu ghi số N là số phần tử của dãy. Dòng sau ghi N số của dãy. N không quá 100, các số trong dãy đều nguyên dương và không quá 1000.

**Kết quả:** Với mỗi bộ test ghi trên hai dòng :

- Dòng thứ nhất là giá trị số lớn nhất
- Dòng thứ 2 lần lượt là các vị trí trong dãy có giá trị lớn nhất (chỉ số tính từ 0).

### BÀI J: ĐẾM SỐ LẦN XUẤT HIỆN

Cho dãy số A có n phần tử chỉ bao gồm các số nguyên dương (không quá  $10^5$ ). Hãy đếm xem mỗi số xuất hiện bao nhiêu lần.

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: dòng đầu ghi số n (không quá 100); dòng tiếp theo ghi n số của dãy.

**Kết quả:** Với mỗi bộ test ghi ra thứ tự bộ test, sau đó lần lượt là các số nguyên tố trong dãy **theo thứ tự xuất hiện trong dãy** và số lần xuất hiện của nó.

Input	Output
1 10 1 7 2 8 3 3 2 1 3 2	Test 1: 1 xuất hiện 2 lần 7 xuất hiện 1 lần 2 xuất hiện 3 lần 8 xuất hiện 1 lần 3 xuất hiện 3 lần

### BÀI K: ĐẾM CÁC SỐ NGUYÊN TỐ TRONG DÃY

Cho dãy số A có n phần tử chỉ bao gồm các số nguyên dương (không quá  $10^5$ ). Hãy xác định các số nguyên tố trong dãy và đếm xem mỗi số xuất hiện bao nhiêu lần.

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test. Với mỗi bộ test: dòng đầu ghi số n (không quá 100); dòng tiếp theo ghi n số của dãy. **Kết quả:** Với mỗi bộ test ghi ra thứ tự bộ test, sau đó lần lượt là các số nguyên tố trong dãy theo thứ tự từ nhỏ đến lớn và số lần xuất hiện của nó.

Input	Output
1 10 1 7 2 8 3 3 2 1 3 2	Test 1: 2 xuất hiện 3 lần 3 xuất hiện 3 lần 7 xuất hiện 1 lần

### BÀI L: LIỆT KÊ VÀ ĐẾM

Cho một dãy các số nguyên dương không quá 9 chữ số, mỗi số cách nhau vài khoảng trống, có thể xuống dòng. Hãy tìm các số không giảm (các chữ số theo thứ tự từ trái qua phải tạo thành dãy không giảm) và đếm số lần xuất hiện của các số đó.

**Dữ liệu vào:** Gồm các số nguyên dương không quá 9 chữ số. Không quá 100000 số.

**Kết quả** Ghi ra các số không giảm kèm theo số lần xuất hiện. Các số được liệt kê theo thứ tự sắp xếp số lần xuất hiện giảm dần. Các số có số lần xuất hiện bằng nhau thì số nào xuất hiện trước in ra trước.

**Ví dụ:**

Input	Output
123 321 23456 123 123 23456 3523 123 321 4567 8988 78 7654 9899 3456 123 678 999 78 3456 987654321 4546 63543 4656 13432 4563 123471 659837 454945 34355 9087 9977 98534 3456 23134	123 5 23456 2 78 2 4567 1 3456 1 678 1 999 1

### BÀI M: SỐ CẶP BẰNG NHAU TRONG DÃY

Viết chương trình đếm các cặp số bằng nhau liên tiếp trong dãy số nguyên.

**Dữ liệu vào:** Dòng đầu tiên ghi số bộ test. Mỗi bộ test có hai dòng:

- Dòng đầu ghi số phần tử của dãy, không quá 30
- Dòng tiếp theo ghi các phần tử của dãy, mỗi phần tử cách nhau một khoảng trống. Các phần tử không quá 100.

**Kết quả:** Mỗi bộ test viết ra trên một dòng giá trị tổng chữ số tương ứng

**Ví dụ:**

Input	Output
2 4 1 3 3 4 12 1 2 3 3 3 3 4 4 5 5 5 1	1 6

### BÀI N: SỐ XUẤT HIỆN NHIỀU LẦN NHẤT TRONG DÃY

Cho một dãy số nguyên dương không quá 100 phần tử, các giá trị trong dãy không quá 30000. Hãy xác định xem số nào là số xuất hiện nhiều lần nhất trong dãy. Chú ý: trong trường hợp nhiều số khác nhau cùng xuất hiện số lần bằng nhau và là lớn nhất thì in ra tất cả các số đó theo thứ tự xuất hiện trong dãy ban đầu.

**Dữ liệu vào:** Dòng đầu là số bộ test, không quá 20. Mỗi bộ test gồm hai dòng. Dòng đầu ghi số phần tử của dãy, dòng tiếp theo ghi các phần tử của dãy.

**Kết quả:** Với mỗi bộ test, đưa ra số xuất hiện nhiều lần nhất trong dãy đã cho.

**Ví dụ:**

Input	Output
2 10 1 2 3 1 2 3 1 2 3 1 10 1 2 3 4 5 6 7 8 9 0	1 1 2 3 4 5 6 7 8 9 0

### BÀI O: TRỘN HAI DÃY VÀ SẮP XẾP

Cho hai dãy số nguyên dương A và B không quá 100 phần tử, các giá trị trong dãy không quá 30000 và số phần tử của hai dãy bằng nhau. Hãy trộn hai dãy với nhau sao cho dãy A được đưa vào các vị trí có chỉ số chẵn, dãy B được đưa vào các vị trí có chỉ số lẻ. Đồng thời, dãy A được sắp xếp tăng dần, còn dãy B được sắp xếp giảm dần. (Chú ý: chỉ số tính từ 0)

**Dữ liệu vào:** Dòng 1 ghi số bộ test. Với mỗi bộ test: dòng đầu tiên ghi số n. Dòng tiếp theo ghi n số nguyên dương của dãy A. Dòng tiếp theo ghi n số nguyên dương của dãy B

**Kết quả:** Với mỗi bộ test, đưa ra thứ tự bộ test và dãy kết quả.

**Ví dụ:**

Input	Output
2 5 1 2 3 1 2 3 1 2 3 1 4 4 2 7 1 5 6 2 8	Test 1: 1 3 1 3 2 2 2 1 3 1 Test 2: 1 8 2 6 4 5 7 2