

KIỂM TRA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
PHẦN 1: LẬP TRÌNH CƠ BẢN VỚI NGÔN NGỮ JAVA

BÀI A: ƯỚC SỐ CHUNG LỚN NHẤT VÀ BỘI SỐ CHUNG NHỎ NHẤT

Viết chương trình tìm ước số chung lớn nhất và bội số chung nhỏ nhất của hai số nguyên dương a,b.

Dữ liệu vào: Dòng đầu ghi số bộ test. Mỗi bộ test ghi trên một dòng 2 số nguyên a và b không quá 9 chữ số.

Kết quả: Mỗi bộ test ghi trên 1 dòng, lần lượt là USCLN, sau đó đến BSCNN.

Ví dụ:

Input	Output
2	2 204
12 34	2 3503326
1234 5678	

BÀI B: LOẠI BỎ HÀNG VÀ CỘT CÓ TỔNG LỚN NHẤT RA KHỎI MA TRẬN

Cho một ma trận cấp $M \times N$ chỉ bao gồm các số nguyên dương không quá 1000. Hãy viết chương trình trước hết loại bỏ hàng có tổng lớn nhất sau đó tính toán lại để loại tiếp cột có tổng lớn nhất ra khỏi ma trận.

Dữ liệu vào

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng hai số N và M lần lượt là số hàng và số cột của ma trận ($2 < N, M < 10$). Tiếp theo là N dòng ghi các số của ma trận.

Kết quả

Mỗi bộ test viết ra thứ tự bộ test, sau đó là ma trận sau khi tính toán.

Ví dụ

Input	Output
1	Test 1:
3 3	1 4
1 2 4	3 0
3 4 0	
6 3 5	

BÀI C: ĐỔI TIỀN

Tại ngân hàng có các mệnh giá bằng 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, số lượng tờ tiền mỗi mệnh giá là không hạn chế. Một người cần đổi số tiền có giá trị bằng N. Hãy xác định xem số tờ tiền ít nhất sau khi đổi là bao nhiêu?

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 50$).

Mỗi test gồm 1 số nguyên N ($1 \leq N \leq 100\,000$).

Output:

Với mỗi test, in ra đáp án trên một dòng.

Test ví dụ:

Input	Output
2	2
70	3
121	

Giải thích test 2: $121 = 100 + 20 + 1$

BÀI D: 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	Test 1:
5	1 3 1 3 2 2 2 1 3 1
1 2 3 1 2	Test 2:
3 1 2 3 1	1 8 2 6 4 5 7 2
4	
4 2 7 1	
5 6 2 8	

BÀI E: SỐ THUẦN NGUYÊN TỐ

Một số được coi là thuần nguyên tố nếu nó là số nguyên tố, tất cả các chữ số là nguyên tố và tổng chữ số của nó cũng là một số nguyên tố. Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số thuần nguyên tố. **Dữ liệu vào:** Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số. **Kết quả:** Mỗi bộ test viết ra số lượng các số thuần nguyên tố tương ứng.

Input	Ouput
2	1
23 199	15
2345 6789	

BÀI F: SỐ LA MÃ

Bảng chữ số La Mã bao gồm các chữ cái với ý nghĩa I=1; V=5; X=10; L=50; C=100; D=500; M=1000. Một số quy tắc viết các số La Mã như sau:

- Tính từ trái sang phải giá trị của các chữ số và nhóm chữ số giảm dần.
- I chỉ có thể đứng trước V hoặc X, X chỉ có thể đứng trước L hoặc C, C chỉ có thể đứng trước D hoặc M.
- Các chữ cái I, X, C, M, không được lặp lại quá ba lần liên tiếp; các chữ cái V, L, D không được lặp lại quá một lần liên tiếp.

Bài toán đặt ra là cho một xâu ký tự mô tả **đúng** một số La Mã. Hãy tính giá trị thập phân của số đó

Dữ liệu vào: Dòng đầu ghi số bộ test. Mỗi bộ test ghi trên một dòng dãy ký tự số La Mã.

Kết quả: Với mỗi bộ test ghi ra kết quả tương ứng

Ví dụ:

Input	Output
3	19
XIX	600
DC	400
CD	

BÀI G: SỐ XUẤT HIỆN NHIỀU NHẤT TRONG DÃY

Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là hãy tìm một số có số lần xuất hiện nhiều nhất trong dãy, và số lần phải lớn hơn N/2.

Dữ liệu vào:

- Dòng đầu tiên là số lượng bộ test T ($T \leq 10$).
- Mỗi test gồm số nguyên N ($1 \leq N \leq 10^5$), số lượng phần tử trong dãy số ban đầu.

- Dòng tiếp theo gồm N số nguyên $A[i]$ ($1 \leq A[i] \leq 10^6$).

Kết quả: Với mỗi test in ra đáp án của bài toán trên một dòng. Nếu không tìm được đáp án, in ra “NO”.

Ví dụ:

Input:	Output
2	4
9	NO
3 3 4 2 4 4 2 4 4	
8	
3 3 4 2 4 4 2 4	

BÀI H: DÃY CON LIÊN TIẾP CÓ TỔNG LỚN NHẤT

Cho dãy số $A[]$ gồm có N phần tử. Nhiệm vụ của bạn là hãy một dãy con liên tiếp sao cho tổng các phần tử của chúng là lớn nhất.

Dữ liệu vào: Dòng đầu tiên là số lượng bộ test T ($T \leq 10$).

Mỗi test gồm số nguyên N ($1 \leq N \leq 10^5$), là số lượng phần tử trong dãy số ban đầu.

Dòng tiếp theo gồm N số nguyên $A[i]$ ($-10^9 \leq A[i] \leq 10^9$).

Output:

Với mỗi test, in ra một số nguyên là đáp án của bài toán trên một dòng.

Test ví dụ:

Input	Output
2	7
8	15
-2 -3 4 -1 -2 1 5 -3	
5	
1 2 3 4 5	

Giải thích test 1: $4 + (-1) + (-2) + 1 + 5 = 7$

BÀI I: XÂU ĐỐI XỨNG

Cho trước một chuỗi S. Bạn hãy thay đổi đúng 1 ký tự (và bắt buộc phải thay đổi 1 ký tự), sao cho chuỗi mới thu được là một chuỗi đối xứng.

Dữ liệu vào: Dòng đầu tiên là số lượng bộ test T ($T \leq 20$). Mỗi test ghi trên một dòng 1 chuỗi S có độ dài không quá 20 ký tự.

Kết quả: Với mỗi test, in ra “YES” nếu có thể biến đổi chuỗi S thành chuỗi đối xứng, in ra “NO” trong trường hợp ngược lại.

Ví dụ:

Input:	Output
3	YES
abccaa	NO
abbcca	YES
abcda	

BÀI J: TRỘN XÂU

Cho hai chuỗi ký tự S1 và S2 với độ dài N và chỉ chứa các ký tự từ A đến H. Chúng ta thực hiện thao tác như sau:

- Bước đầu tiên tạo S12 bằng cách lấy các ký tự lần lượt trong S1 và S2 từ trái sang phải, lấy một ký tự trong S2 trước, sau đó đến 1 ký tự trong S1 và cứ như vậy. Ví dụ S1 = ABCHAD, S2 = DEFDAC thì S12 = DAEBFCDHAACD
- Sau đó ta lại lấy nửa bên trái của S12 thành S1 mới, nửa bên phải thành S2 mới. Trong ví dụ trên S1 mới là DAEBFC, S2 mới là DHAACD. Rồi lại tiếp tục như vậy trong các bước tiếp theo.

Cho trước một chuỗi S có độ dài $2 \cdot N$. Bài toán đặt ra là liệu có thể tạo ra chuỗi S sau một số lần lặp hay không.

Dữ liệu vào

Có nhiều bộ test, mỗi bộ test có bốn dòng. Dòng đầu ghi số N không quá 100. Dòng thứ 2 ghi S1, dòng thứ 3 ghi S2. Dòng cuối ghi chuỗi S. Input kết thúc với một dòng ghi số 0.

Kết quả

Ghi ra số bước lặp cần thiết. Nếu không thể tìm được thì ghi ra -1.

Ví dụ

Input	Output
4	2
AHAH	-1
HAHA	
HHAAAAHH	
3	
CDE	
CDE	
EEDDCC	
0	