

Học sinh tạo thư mục là họ tên và lớp (viết liền không dấu, VD: leebook\_11T8), lưu các bài làm với tên tương ứng bai1.???, bai2.???, bai3.???, bai4.???, bai5.??? vào thư mục vừa tạo (dấu ??? được thay bằng phần mở rộng của ngôn ngữ lập trình dùng để viết chương trình, VD: bai1.cpp, bai1.pas, bai1.py, ...).

Hãy lập trình giải các bài toán bên dưới.

### Bài 1. (4 điểm) PHÉP CỘNG KHẢ THI??

Bạn được cho ba số nguyên  $a, b$ , và  $c$ .

**Yêu cầu:** Xác định xem một trong số chúng có phải là tổng của hai số còn lại hay không.

**Dữ liệu vào:** Từ FILE "bai1.in". **Bao gồm:** Dòng đầu tiên chứa một số nguyên  $t$  ( $1 \leq t \leq 9261$ ) — số lượng bài kiểm tra. Mô tả của mỗi bài kiểm tra bao gồm ba số nguyên  $a, b, c$  ( $0 \leq a, b, c \leq 20$ ).

**Dữ liệu ra:** Ra FILE "bai1.out". **Bao gồm:** Đối với mỗi bài kiểm tra, xuất "YES" nếu một trong các số là tổng của hai số còn lại, và "NO" nếu không.

**Ràng buộc:** Ràng buộc chung:  $1 \leq t \leq 9261; 0 \leq a, b, c \leq 20$ .

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

Ví dụ:

bai1.in	bai1.out
7	YES
1 4 3	NO
2 5 8	YES
9 11 20	YES
0 0 0	NO
20 20 20	NO
4 12 3	NO
15 7 8	YES

**Ghi chú:** Trong bài kiểm tra đầu tiên,  $1 + 3 = 4$ . Trong bài kiểm tra thứ hai, không có số nào là tổng của hai số còn lại. Trong bài kiểm tra thứ ba,  $9 + 11 = 20$ .

### Bài 2. (4 điểm) TRUY HỒI GIÁ TRỊ

Leebook đã đoán ba số nguyên dương  $a, b$  và  $c$ . Anh ấy giữ những số này trong bí mật, nhưng anh ấy đã viết bốn số lên bảng theo thứ tự tùy ý — tổng của chúng theo cặp (ba số) và tổng của cả ba số (một số). Vì vậy, có bốn số trên bảng theo thứ tự ngẫu nhiên:  $a + b$ ,  $a + c$ ,  $b + c$  và  $a + b + c$ . Bạn phải đoán ba số  $a, b$  và  $c$  bằng cách sử dụng các số đã cho. In ra ba số nguyên đã đoán theo bất kỳ thứ tự nào. Lưu ý rằng một số số đã cho  $a, b$  và  $c$  có thể bằng nhau (cũng có thể  $a = b = c$ ).

**Dữ liệu vào:** Từ FILE "bai2.in". **Bao gồm:** Dòng duy nhất của đầu vào chứa bốn số nguyên dương  $x_1, x_2, x_3, x_4$  ( $2 \leq x_i \leq 10^9$ ) — các số được viết trên bảng theo thứ tự ngẫu nhiên. Đảm bảo rằng câu trả lời tồn tại cho số đã cho  $x_1, x_2, x_3, x_4$ .

**Dữ liệu ra:** Ra FILE "bai2.out". **Bao gồm:** In ra các số nguyên dương  $a, b$  và  $c$  sao cho bốn số

được viết trên bảng là các giá trị  $a + b$ ,  $a + c$ ,  $b + c$  và  $a + b + c$  được viết theo một thứ tự nào đó. In ra  $a$ ,  $b$  và  $c$  theo bất kỳ thứ tự nào. Nếu có nhiều câu trả lời, bạn có thể in bất kỳ câu nào. Đảm bảo rằng câu trả lời tồn tại.

**Ràng buộc:** Ràng buộc chung:  $1 \leq t \leq 100$ .

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 256 MB

Ví dụ:

bai2.in	bai2.out
3 6 5 4	2 1 3

bai2.in	bai2.out
40 40 40 60	20 20 20

bai2.in	bai2.out
201 101 101 200	1 100 100

### Bài 3. (4 điểm) CHUỖI VUÔNG

Một chuỗi được gọi là hình vuông nếu nó là một chuỗi nào đó được viết hai lần liên tiếp. Ví dụ, các chuỗi "aa", "abcabc", "abab" và "baabaa" là hình vuông. Nhưng các chuỗi "aaa", "abaaab" và "abcdabc" thì không phải là hình vuông. Cho một chuỗi  $s$  xác định xem nó có phải là hình vuông hay không.

**Dữ liệu vào:** Từ FILE "bai3.in". **Bao gồm:** Dòng đầu tiên của dữ liệu đầu vào chứa một số nguyên  $t$  ( $1 \leq t \leq 100$ —số lượng các trường hợp kiểm tra. Tiếp theo là  $t$  dòng, mỗi dòng chứa một mô tả của một trường hợp kiểm tra. Các chuỗi được cho chỉ bao gồm các chữ cái Latin viết thường và có độ dài từ 1 đến 100 bao gồm.

**Dữ liệu ra:** Ra FILE "bai3.out". **Bao gồm:** Đối với mỗi trường hợp kiểm tra, xuất ra trên một dòng riêng biệt: "YES" nếu chuỗi trong trường hợp kiểm tra tương ứng là hình vuông, "NO" nếu không.

**Ràng buộc:** Ràng buộc chung:  $1 \leq t \leq 100$ .

**Giới hạn thời gian, bộ nhớ:** 1 giây (1000 ms); 512 MB

Ví dụ:

bai3.in	bai3.out
10	NO
a	YES
aa	NO
aaa	YES
aaaa	YES
abab	YES
abcabc	NO
abacaba	NO
xyxy	NO
xyyx	YES
xyxy	

### Bài 4. (4 điểm) MẢNG SỐ NGUYÊN KHÔNG GIẢM

Bạn được cho hai mảng số nguyên  $a$  và  $b$  đều có độ dài  $n$ . Bạn có thể chọn bất kỳ tập con chỉ số nào và hoán đổi các phần tử ở những vị trí đó (tức là thực hiện  $\text{swap}(a_i, b_i)$  cho mỗi  $i$  trong tập con). Một tập con chỉ số được gọi là **tốt** nếu sau khi hoán đổi, cả hai mảng đều được sắp xếp theo thứ tự không giảm. Nhiệm vụ của bạn là tính số lượng tập con tốt. Vì kết quả có thể rất lớn, hãy in ra kết

quả theo modulo 998244353

**Dữ liệu vào: Từ FILE "bai4.in". Bao gồm:**

Dòng đầu tiên chứa một số nguyên  $t$  ( $1 \leq t \leq 500$ ) — số lượng bộ test.

Dòng đầu tiên của mỗi bộ test chứa một số nguyên  $n$  ( $1 \leq n \leq 100$ ).

Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 1000$ ).

Dòng thứ ba chứa  $n$  số nguyên  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 1000$ ).

Ràng buộc thêm: luôn tồn tại ít nhất một tập con tốt.

**Dữ liệu ra: Ra FILE "bai4.out". Bao gồm:** Với mỗi bộ test, in ra một số nguyên duy nhất — số lượng tập con tốt, lấy modulo 998244353.

**Giới hạn thời gian, bộ nhớ: 2 giây (2000 ms); 256 MB**

**Ví dụ:**

bai4.in	bai4.out
3	2
3	2
2 1 4	8
1 3 2	
1	
4	
4	
5	
2 3 3 4 4	
1 1 3 5 6	

**Giải thích:**

- ★ Trong ví dụ đầu tiên có 2 tập con tốt:  $\{1; 3\}$  và  $\{2\}$ .
- ★ Trong ví dụ thứ hai, có 2 tập con tốt:  $\{1\}$  và  $\{\}$ .
- ★ Trong ví dụ thứ ba, có 8 tập con tốt:  $\{1; 2; 3; 4; 5\}$ ;  $\{1; 2; 3\}$ ;  $\{1; 2; 4; 5\}$ ;  $\{1; 2\}$ ;  $\{3; 4; 5\}$ ;  $\{3\}$ ;  $\{4; 5\}$  và  $\{\}$ .

### Bài 5. (4 điểm) HÃY CHỌN GIÁ ĐÚNG??

Hãy tưởng tượng bạn là chủ một cửa hàng. Trước khi bắt đầu một mùa mới, bạn quyết định thanh lý hết hàng tồn kho, nên bạn quyết định tổ chức một đợt giảm giá tổng lực. Bạn có  $n$  món hàng khác nhau trong cửa hàng: món hàng thứ  $i$  có giá  $c_i$  đồng. Mỗi món đều có một mức giá tương ứng  $c_i$ . Bạn quyết định giảm giá theo kiểu: "chúng ta sẽ chia tất cả giá thành  $x$  lần." Nói chính xác hơn, bạn chọn một hệ số chung  $x$ , và trong đợt giảm giá, món hàng thứ  $i$  sẽ có giá  $\left\lceil \frac{c_i}{x} \right\rceil$  đồng (ở đây  $\lceil y \rceil$  là làm tròn lên). Để tránh khách hàng bị rối, bạn cần dán lại mức giá mới cho tất cả món hàng, nhưng việc in mức mới rất tốn kém. Cụ thể, mỗi mức in ra sẽ tiêu tốn của bạn  $y$  đồng. Vậy nên bạn nảy ra một ý tưởng tuyệt vời — tại sao không tận dụng các mức cũ rồi dán lại cho những món khác? Như vậy, bạn chỉ phải in mức mới cho những món không có mức phù hợp sẵn. Câu hỏi cuối cùng là: bạn nên giảm giá bao nhiêu, hay nói cách khác chọn  $x$  thế nào? Hệ số  $x$  phải là một số **nguyên** lớn hơn hẳn 1 sao cho tổng thu nhập là cao nhất. Tổng thu nhập được tính bằng tổng giá trị các món hàng trừ đi chi phí in mức mới. Hãy xác định tổng thu nhập tối đa có thể đạt được.

**Dữ liệu vào: Từ FILE "bai5.in". Bao gồm:**

Dòng đầu tiên chứa một số nguyên duy nhất  $t$  ( $1 \leq t \leq 10$ ) — số bộ test.

Dòng đầu mỗi bộ test gồm hai số nguyên  $n$  và  $y$  ( $1 \leq n \leq 2 \cdot 10^5$ ;  $1 \leq y \leq 10^9$ ) — số món hàng và chi phí in một mức giá.

Dòng tiếp theo gồm  $n$  số nguyên  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 2 \cdot 10^5$ ) — giá ban đầu của từng món hàng.

**Dữ liệu ra: Ra FILE "bai5.out". Bao gồm:** Với mỗi bộ test, in ra một số nguyên duy nhất — tổng thu nhập tối đa.

**Giới hạn thời gian, bộ nhớ: 2 giây (2000 ms); 256 MB**

**Một số ví dụ:**

<b>bai5.in</b>	<b>bai5.out</b>
4	31
5 51	-2999999937
50 150 50 148 150	-162755
3 1000000000	3
42 42 42	
10 54321	
1 8088 45 1 73 1 9198 4991 1 83	
3 100	
1 1 1	

**Lý giải:**

- Trong bộ test đầu tiên, chọn  $x = 3$  là tối ưu. Giá mới của các món sẽ là  $[17, 50, 17, 50, 50]$ . Lúc này, ta có thể tận dụng hai mức cũ với giá 50, và phải in mới ba mức cho các món 17, 17, và 50. Kết quả thu nhập là  $17 + 50 + 17 + 50 + 50 - 51 \cdot 3 = 31$ .
- Ở bộ test thứ hai, chọn  $x = 2$  là tốt nhất. Giá mới sẽ là  $[21, 21, 21]$ , và ta phải in 3 mức mới.
- Bộ test thứ ba, chọn  $x = 111$  là tối ưu.
- Bộ test thứ tư, chọn  $x = 2$  là hợp lý nhất. Giá mới giống hết giá cũ, nên không cần in mức mới nào.

———— **HẾT** ————