

### 1. nummax.cpp

Cho số nguyên không âm  $x$ , tìm số nguyên  $n$  lớn nhất thỏa mãn:  $0 \leq n \leq x$  và tất cả các chữ số trong biểu diễn thập phân của  $n$  đều giống nhau.

**Dữ liệu:** một số nguyên  $x$  ( $0 \leq x \leq 10^{18}$ )

**Kết quả:** Ghi ra số  $n$  tìm được.

**Ví dụ:**

INPUT	OUTPUT
2	2
56789	55555
43210	33333
1000000000000000000	999999999999999999

### 2. game1.cpp

ANDY và BOB đều rất thích chơi những trò chơi liên quan đến dãy số. Hai bạn đã cùng nhau nghĩ ra một trò chơi đơn giản mà vô cùng thú vị. Luật chơi như sau:

- Hai bạn chọn ra một dãy số gồm  $n$  số nguyên dương phân biệt, rồi lần lượt thay phiên nhau thực hiện lượt chơi của mình.
- Khi tới lượt, người chơi chỉ ra số lớn nhất trong dãy số hiện tại và xóa bỏ số đó cùng với tất cả những số khác nằm ở vị trí bên phải số này khỏi dãy. Sau đó, người chơi kết thúc lượt của mình và chuyển lượt cho người còn lại. Trò chơi sẽ tiếp tục với dãy số sau khi đã thực hiện thao tác xóa.

- Nếu người chơi không thể thực hiện thao tác nào trong lượt của mình thì sẽ bị xử thua.

BOB luôn là người xung phong chơi trước. Hai bạn mãi mê chơi với nhau  $t$  ván đến nỗi quên cả ghi lại kết quả sau mỗi ván. Vì vậy, bạn hãy thay ANDY và BOB ghi lại tên người thắng cuộc sau mỗi ván nhé.

**Yêu cầu:** Hãy giúp ANDY và BOB thực hiện yêu cầu trên.

**Dữ liệu:**

- Dòng đầu chứa một số nguyên dương  $t$ .
- Tiếp theo là  $t$  ván chơi, mỗi ván mô tả bởi hai dòng:
- Dòng thứ nhất chứa một số nguyên dương  $n$ .
- Dòng thứ hai chứa  $n$  số nguyên dương phân biệt  $a_1, a_2, \dots, a_n$  cách nhau bởi một khoảng trống.

**Kết quả:** Với mỗi ván chơi, in ra tên người thắng cuộc (ANDY hoặc BOB) theo từng dòng.

**Ví dụ:**

INPUT	OUTPUT
2	ANDY
5	BOB
5 2 6 3 4	
2	
3 1	

**Ràng buộc:**

- Có 35% số test ứng với 35% số điểm có:

$$1 \leq t \leq 10.$$

$$1 \leq n \leq 1000.$$

$$1 \leq a_i \leq 10^5.$$

Tổng  $n$  của tất cả ván chơi không vượt quá 1000.

- Có 65% số test còn lại ứng với 65% số điểm có:

$$1 \leq t \leq 100.$$

$$1 \leq n \leq 10^5.$$

$$1 \leq a_i \leq 10^9.$$

### 3. spray.cpp

Vườn nhà An có trồng  $n$  cây, cây thứ  $i$  có độ tươi tốt là  $a_i$  và khả năng tươi tăng trưởng  $b_i$ .

Hôm nay, An định dùng  $L$  lít nước để tưới cho các cây trong vườn. Với mỗi lít nước tưới vào một cây thứ  $i$ , độ tươi tốt của cây sẽ được tăng lên  $b_i$ . Ngoài ra số lít nước tưới vào mỗi cây phải là một số nguyên.

Vẻ đẹp của khu vườn là độ tươi tốt bé nhất trong số  $n$  cây trong vườn. Hãy giúp An tưới cây sao cho vẻ đẹp của khu vườn là lớn nhất có thể.

**Dữ liệu:**

- Dòng 1: Chứa hai số nguyên dương  $n, L$  ( $1 \leq n \leq 10^5, 1 \leq L \leq 10^9$ ) – số cây trong vườn và số lít nước dùng để tưới cây.
- $N$  dòng tiếp theo, dòng thứ  $i$  gồm hai số nguyên  $a_i$  và  $b_i$  ( $\forall i: 1 \leq a_i, b_i \leq 10^4$ ) – độ tươi tốt và khả năng tăng trưởng của cây thứ  $i$ .

**Kết quả:** Ghi ra vẻ đẹp lớn nhất của khu vườn với cách tưới tối ưu.

*Ví dụ:*

INPUT	OUTPUT	Giải thích
3 5 1 5 6 2 3 3	8	Tưới 2 lít nước vào cây thứ nhất. 1 lít nước vào cây thứ hai và 2 lít nước vào cây thứ 3. Khi đó, độ tươi tốt của các cây lần lượt là [11, 8, 9] và vẻ đẹp của khu vườn là 8.

**Ràng buộc:**

- Có 30%:  $n, L \leq 1000$ .
- Có 30%:  $L \leq 10^5$
- 40% không có ràng buộc thêm

### 4. addnum.cpp

Cho dãy số nguyên  $A = (a_1, a_2, \dots, a_n)$  và số nguyên dương  $X$ . Trung vị của dãy gồm  $m$  phần tử là phần tử thứ  $(m+1)/2$  sau khi sắp xếp dãy đó không giảm.

**Yêu cầu:** Cho biết cần thêm ít nhất bao nhiêu số vào dãy  $A$  để  $X$  là phần tử trung vị của dãy  $A$ .

**Dữ liệu:**

- Dòng 1: Chứa hai số nguyên dương  $n, x$  ( $1 \leq n \leq 10^6, 1 \leq X \leq 10^9$ )
- Dòng 2: Chứa  $n$  số nguyên dương  $a_i$  ( $\forall i: a_i \leq 10^9$ )

**Kết quả:** Ghi ra một số nguyên là số lượng số ít nhất cần thêm vào

*Ví dụ:*

INPUT	OUTPUT	Giải thích
-------	--------	------------

3 2	1	Thêm số 1
10 10 2		

## 5. open.cpp

Một coder trẻ với mật danh Hacker đã chiến thắng giải đấu Open và giành được chiếc laptop độc quyền của nhà tài trợ. Chiếc laptop này có cài đặt sẵn một trình duyệt internet do công ty phát triển với các thử nghiệm hết sức độc đáo. Tuy nhiên trình duyệt bị giới hạn khi chỉ có thể mở được nhiều nhất  $k$  cửa sổ và tab thứ  $i$  trong mỗi cửa sổ chiếm  $i$  megabytes trong bộ nhớ. Hacker cho biết chiếc laptop mới có bộ nhớ  $m$  megabytes.

**Yêu cầu:** Bạn hãy giúp Hacker tính xem cậu ta có thể mở được nhiều nhất là bao nhiêu tab.

**Dữ liệu:**

- Dòng 1: Chứa số nguyên dương  $T$  ( $1 \leq T \leq 10^5$ ) – số test.
- Dòng thứ  $i + 1$  ( $1 \leq i \leq T$ ) gồm hai số nguyên  $m, k$  ( $1 \leq m \leq 10^{18}$ ,  $1 \leq k \leq 10^9$ ) là tổng bộ nhớ của laptop và số cửa sổ nhiều nhất có thể mở.

**Kết quả:** Gồm  $T$  dòng, mỗi dòng gồm một số nguyên duy nhất là số tab nhiều nhất mà Hacker có thể mở.

**Ví dụ:**

INPUT	OUTPUT
2	10
23 3	2
2 3	

## 6. dcandy.cpp

Halloween đã đến, các bạn nhỏ đan giỏ của mình để xin kẹo. Khi đến nhà An, An bảo các bạn nhỏ xếp thành một hàng và kiểm tra số kẹo trong giỏ của các bạn nhỏ, An đếm được lần lượt các bạn nhỏ có  $A_1, A_2, \dots, A_N$  kẹo. Để làm khó các bạn nhỏ, An đã chuẩn bị  $M$  túi kẹo, xếp thành một chồng, túi kẹo thứ  $i$  có  $B_i$  kẹo. An cho các bạn nhỏ lần lượt lên nhận kẹo với quy tắc:

- Mỗi bạn có thể nhận không quá một túi kẹo. Chỉ có thể nhận túi kẹo trên đầu của chồng kẹo.
- Có thể bỏ đi một số túi kẹo trên đầu, và tất nhiên khi bỏ đi, An sẽ thu túi kẹo đó lại.

**Yêu cầu:** Bạn hãy lập kế hoạch nhận kẹo cho các bạn nhỏ sao cho sau khi nhận kẹo, số kẹo của bạn có ít kẹo nhất là lớn nhất.

**Dữ liệu:**

- Dòng 1: gồm số nguyên dương  $N$  là số lượng bạn nhỏ đi xin kẹo.
- Dòng 2: gồm  $N$  số nguyên dương là số kẹo ban đầu của  $N$  bạn nhỏ.
- Dòng 3: gồm số nguyên dương  $M$  là số túi kẹo của An.
- Dòng 4: gồm  $M$  số nguyên dương là số kẹo trong  $M$  túi kẹo của An.

**Kết quả:** Ghi ra một số nguyên duy nhất là kết quả của bài toán.

**Ví dụ:**

INPUT	OUTPUT
6	6

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

**Giải thích:**

- Bạn nhỏ 1: bỏ qua gói kẹo đầu tiên, lấy gói kẹo thứ 2 => Có  $2+5=7$  kẹo
- Bạn nhỏ 2: không lấy gói kẹo nào => Có 8 kẹo
- Bạn nhỏ 3: lấy gói kẹo thứ 3 => Có  $4+3=7$  kẹo
- Bạn nhỏ 4: bỏ qua gói kẹo thứ 4 và thứ 5, lấy gói kẹo thứ 6 => Có  $2+4=6$  kẹo
- Bạn nhỏ 5: không lấy gói kẹo nào => Có 7 kẹo
- Bạn nhỏ 6: không lấy gói kẹo nào => Có 6 kẹo

Vậy bạn có ít kẹo nhất là 6 kẹo.

**Ràng buộc:**  $A_i, B_i \leq 10^9$

- Có 30% số test có  $N, M \leq 10$ ;
- 30% số test có  $N, M \leq 1000$ ;
- 40% số test còn lại có  $N, M \leq 10^5$ .