



B. George and Round

Link submit: <http://codeforces.com/problemset/problem/387/B>

Solution:

C/C++: <http://ideone.com/LQIDK5>

Python: <https://ideone.com/37iKjU>

Java: <https://ideone.com/nVUgCe>

Tóm tắt đề: George chuẩn bị một kỳ thi cho Codeforces. Để có được một kỳ thi hay thì cần phải có những problem tốt. Cho một kỳ thi của Codeforces và số lượng problem tốt cần có.

Cho số lượng problem hiện tại George đang có, bạn cần kiểm tra xem với những problem mà George đang có thì có cần thêm bao nhiêu problem nữa để tạo ra được một kỳ thi Codeforces tốt.

Mỗi problem được đại diện là 1 con số, số càng lớn thì độ phức tạp càng lớn.

Input

Dòng đầu tiên chứa 2 số n và m ($1 \leq n, m \leq 3000$) - n là cho số problem tối thiểu cần có trong kỳ thi, m là số problem hiện có.

Dòng thứ 2 chứa n số nguyên ($1 \leq a_1 < a_2 < \dots < a_n \leq 10^6$) ứng với độ phức tạp cần có của các problem trong kỳ thi.

Dòng thứ 3 chứa m số nguyên ($1 \leq b_1 \leq b_2 \leq \dots \leq b_m \leq 10^6$) ứng với độ phức tạp của các problem hiện có.

Output

In ra một con số duy nhất là số lượng bài toán cần chuẩn bị thêm cho kỳ thi nếu các problem của George không đủ để chuẩn bị.

3 5 1 2 3	0
--------------	---

1 2 2 3 3	
-----------	--

Giải thích: Kỳ thi có 3 bài, G có 5 bài.

Kỳ thi cần có 3 bài với cấp độ 1, 2, 3. Trong 5 bài của George đã có đủ các bài với độ phức tạp trên nên không cần thêm bài nào. Xuất ra giá trị 0.

Hướng dẫn giải:

Lần lượt bỏ độ khó của các bài vào vector a (bài toán sẽ ra) và vector b là các bài toán có sẵn.

Có 2 vòng lặp, vòng lặp đầu tiên sẽ chạy qua lần lượt các độ khó của vector a, trong lúc chạy xét xem phần tử nào trong b thỏa mãn có thể ra đề thì tăng biến đếm lên.

Kết quả (n – biến đếm) chính là số bài cần chuẩn bị thêm.

Độ phức tạp: $O(\max(n,m))$ với n và m lần lượt là số lượng bài cần ra trong kỳ thi và số lượng problem hiện có.