

## SỬA HÀNG RÀO (WALL.PAS, WALL.CPP)

Bờm dựng đã dựng xong một hàng rào gồm  $n$  thanh gỗ độ rộng đơn vị ghép liền nhau, đánh số từ 1 đến  $n$ , thanh gỗ thứ  $i$  có độ cao là  $a_i$ . Tuy vậy, vẫn còn thừa  $m$  thanh gỗ nữa chưa sử dụng. Anh ta quyết định dùng  $m$  thanh gỗ còn thừa này để nâng chiều cao của hàng rào vừa dựng xong.  $m$  thanh gỗ này được Bờm xếp lên một chiếc xe thành một chồng, tính từ trên xuống dưới các thanh gỗ được đánh số từ 1 đến  $m$ , thanh gỗ thứ  $i$  có chiều cao là  $b_i$  ( $i = 1, 2, \dots, m$ ). Bờm sẽ kéo chiếc xe này từ đầu hàng rào đến cuối hàng rào (từ thanh số 1 đến thanh số  $n$ ). Mỗi khi đến vị trí của một thanh gỗ anh ta có thể:

- Hoặc bỏ qua không làm gì cả
- Hoặc lấy một thanh gỗ bất kỳ còn lại trên xe ba gác (khi đó tất cả các thanh gỗ ở bên trên nó sẽ bị bỏ ra khỏi xe ba gác và không bao giờ sử dụng nữa) đóng tiếp lên vị trí của thanh gỗ đang đứng, chiều cao mới tại vị trí này sẽ là tổng chiều cao của thanh gỗ cũ và thanh gỗ mới đóng lên.

Hãy xác định độ lớn nhất của hàng rào sau khi nâng chiều cao theo cách nói trên. Độ cao của hàng rào được tính là độ cao của thanh gỗ có độ cao thấp nhất.

**Dữ liệu:** Vào từ file WALL.INP gồm nhiều test ghi liên tiếp nhau. Mỗi test có cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 10^5$ )
- Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^8, i = 1, 2, \dots, n$ )
- Dòng thứ ba chứa số nguyên  $m$  ( $1 \leq m \leq 10^5$ )
- Dòng thứ ba chứa  $m$  số nguyên  $b_1, b_2, \dots, b_m$  ( $1 \leq b_i \leq 10^5, i = 1, 2, \dots, m$ )

**Kết quả:** Ghi ra file WALL.OUT Kết quả các test ghi lần lượt theo cấu trúc:

- Dòng đầu ghi hai số nguyên  $h$  và  $k$  - độ cao lớn nhất của hàng rào và số thanh gỗ được đóng thêm.
- Mỗi dòng trong  $k$  dòng tiếp theo ghi hai số nguyên  $u$  và  $v$  trong đó  $u$  - số hiệu của thanh gỗ được đóng lên và  $v$  - số hiệu của thanh gỗ thừa được dùng để đóng (các cặp  $(u, v)$  đưa ra theo thứ tự tăng dần).

**Ví dụ:**

WALL.INP	WALL.OUT
6	5 3
2 5 4 1 7 5	1 2
7	3 4
2 3 1 3 2 4 6	4 7