

TỔNG ĐOẠN

Cho một dãy số $a_0, a_1, a_2, \dots, a_{n-2}, a_{n-1}$ là các số nguyên $|a_i| \leq 2 \cdot 10^9$. Ban đầu tất cả các số có giá trị 0 và trên dãy số có thể thực hiện hai lệnh sau:

Lệnh cập nhật $S(i, k)$:

Gán giá trị k cho phần tử a_i ($0 \leq i \leq n - 1; |k| \leq 2 \cdot 10^9$).

Lệnh truy vấn $Q(i, j)$:

Cho biết tổng của các số $a_i, a_{i+1}, \dots, a_{j-1}, a_j$ ($0 \leq i \leq j \leq n - 1$).

Yêu cầu: Cho một dãy m lệnh thuộc một trong hai loại trên, hãy trả lời tất cả các lệnh truy vấn.

Dữ liệu: Vào từ file văn bản **SUM.INP**

Dòng 1 chứa hai số nguyên dương $n, m \leq 10^5$.

m dòng tiếp theo, mỗi dòng chứa thông tin về một lệnh, đầu tiên là một ký tự $\in \{S, Q\}$. Nếu ký tự đầu dòng là S , tiếp theo là hai số nguyên i, k cho biết đó là lệnh $S(i, k)$. Nếu ký tự đầu dòng là Q , tiếp theo là hai số nguyên i, j cho biết lệnh $Q(i, j)$.

Kết quả: Ghi ra file văn bản **SUM.OUT**

Tương ứng với mỗi lệnh truy vấn Q trong file dữ liệu, ghi ra trên một dòng một số nguyên là trả lời cho truy vấn đó.

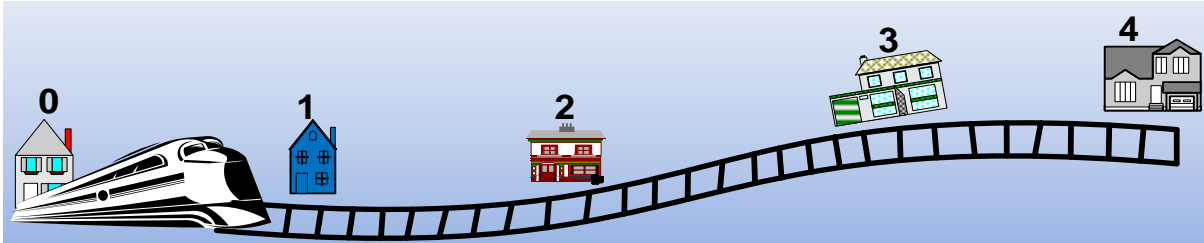
Ví dụ:

SUM.INP	SUM.OUT
5 6	6
S 2 1	18
S 4 5	
Q 2 4	
S 3 6	
S 2 7	
Q 1 4	

VÉ TÀU

Tên chương trình: TICKET.???

Tuyến tàu hỏa tốc hành chất lượng cao có n ga đánh số từ 0 đến $n-1$. Các đăng ký mua vé được gửi đến trung tâm điều độ hành khách và được đáp ứng trong phạm vi có thể theo trình tự xuất hiện. Trên tàu có k chỗ ngồi. Vé sẽ được bán cho hành khách nếu trên đoạn đường yêu cầu còn ghế trống. Khi hành khách



xuống tại một ga nào đó, ghế của người ấy sẽ được coi là trống kể từ ga đó và có thể bán cho người khác có nhu cầu đi. Thông tin để xử lý mỗi vé là x và y – ga lên và xuống của hành khách ($0 \leq x < y \leq n-1$). Nếu còn chỗ, hệ thống sẽ đưa ra thông báo ‘1’ – yêu cầu được đáp ứng, trong trường hợp ngược lại – hệ thống sẽ đưa ra thông báo ‘0’ – hết chỗ.

Yêu cầu: Cho n , k và m – số lượng các yêu cầu đăng ký vé, mỗi yêu cầu được nêu dưới dạng 2 số nguyên x và y . Các yêu cầu được liệt kê theo trình tự xuất hiện. Với mỗi yêu cầu hãy xác định có thể đáp ứng được hay không.

Dữ liệu: Vào từ file văn bản **TICKET.INP**:

- Dòng đầu tiên chứa 3 số nguyên n , k và m ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^4$, $1 \leq m \leq 5 \times 10^5$),
- Mỗi dòng trong m dòng tiếp theo chứa 2 số nguyên x và y .

Kết quả: Đưa ra file văn bản **TICKET.OUT** m dòng, dòng thứ i chứa số 1 nếu yêu cầu thứ i được đáp ứng và đưa ra số 0 trong trường hợp ngược lại.

Ví dụ:

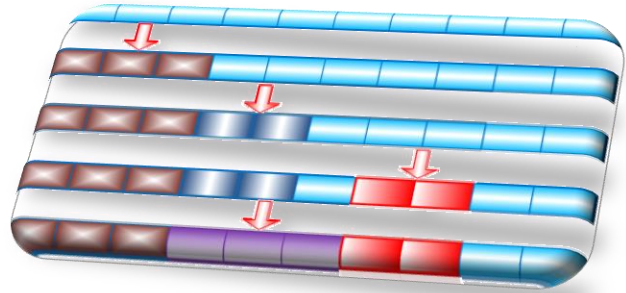
TICKET.INP		
5	2	4
0	4	
1	2	
1	4	
2	4	

TICKET.OUT
1
1
0
1

HỆ ĐIỀU HÀNH

Tên chương trình: OS.???

Đĩa cứng của Steve có M sectors ($1 \leq M \leq 10^9$). Các sectors được đánh số từ 1 đến M . Steve lần lượt cài các hệ điều hành khác nhau theo cách sau: tạo một vùng mới từ sector a_i sector b_i (kể cả b_i), cài tiếp lên đó hệ điều hành mới. Nếu vùng này đè lên dù chỉ một sector của một vùng nào trước đó thì hệ thống đã cài ở vùng đó sẽ bị hỏng, không khai thác tiếp được.



Yêu cầu: Cho M, n – số lần cài đặt ($0 \leq n \leq 100\,000$) và các giá trị a_i, b_i ($i = 1 \div n$). Hãy xác định số Hệ điều hành còn khai thác được.

Dữ liệu: Vào từ file văn bản **OS.INP**:

- Dòng đầu tiên chứa số nguyên M ,
- Dòng thứ 2 chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa 2 số nguyên a_i và b_i .

Các dữ liệu được đưa theo trình tự cài đặt.

Kết quả: Đưa ra file văn bản **OS.OUT** một số nguyên – số hệ điều hành hoạt động được.

Ví dụ:

OS.INP	
10	
4	
1	3
4	5
7	8
4	6

OS.OUT
3