

## Bài 1: Xếp hàng mua vé

Có  $N$  người xếp hàng mua vé. Ta đánh số họ từ 1 đến  $N$  theo thứ tự đứng trong hàng. Thời gian phục vụ bán vé cho người thứ  $i$  là  $t_i$ . Mỗi người cần mua một vé nhưng được quyền mua tối đa 2 vé, vì thế một số người có thể nhờ người đứng ngay trước mình mua hộ. Người thứ  $i$  nhận mua hộ vé cho người thứ  $i+1$  thì thời gian mua vé cho 2 người là  $r_i$ . Tìm phương án sao cho  $N$  người đều có vé với thời gian ít nhất.

**Dữ liệu** vào từ file văn bản **TICK.INP**

- Dòng thứ nhất ghi số nguyên dương  $N$  ( $1 < N \leq 2000$ )
- Dòng thứ hai ghi  $N$  số nguyên dương  $t_1, t_2, \dots, t_n$
- Dòng thứ ba ghi  $N-1$  số  $r_1, r_2, \dots, r_{n-1}$

**Kết quả** ghi ra file **TICK.OUT**

- Dòng thứ nhất ghi tổng thời gian phục vụ bán vé

**Ví dụ**

TICK.INP	TICK.OUT
5	17
2 5 7 8 4	
3 9 10 10	

## Bài 2: Tính tổng

Trên một màn hình lớn, người ta lần lượt cho hiện ra các số của một dãy gồm  $N$  số nguyên không âm  $a_1, a_2, \dots, a_N$  và cứ lặp đi lặp lại như thế (nghĩa là sau khi  $a_i$  xuất hiện vài giây đến lượt  $a_{i+1}$  xuất hiện, số xuất hiện sau  $a_N$  là  $a_1$ ).

**Yêu cầu:** Bạn được đề nghị tính tổng của  $K$  số nguyên liên tiếp xuất hiện trên màn hình bắt đầu từ số nguyên thứ  $B$ .

**Dữ liệu vào:** File **SUM.INP** gồm hai dòng:

- + Dòng đầu tiên ghi ba số nguyên  $N, K$ , và  $B$ ,  $1 \leq N \leq 10^4$ ;  $1 \leq K \leq 2 \cdot 10^4$ ;  $1 \leq B \leq 10^9$ .
- + Trong  $N$  dòng sau, dòng thứ  $i$  chứa số  $a_i$  ( $a_i < 2 \cdot 10^9$ ).

**Dữ liệu ra:** File **SUM.OUT** một số là tổng tìm được.

**Ví dụ:**

SUM.INP
---------

SUM.OUT
---------

5	7	6
1		
2		
3		
4		
5		

18
----

### Bài 3: Xếp các hộp chồng nhau

Để chuẩn bị cho kì thi OLP 2022, các phòng máy tính của trường Chuyên LQĐ được cung cấp một số tiền lớn mua trang thiết bị. Tuy nhiên, sau khi hoàn thành việc lắp đặt, phòng máy ngổn ngang các hộp hình chữ nhật đựng CPU, chuột, màn hình, bàn phím, điều hòa.... Các thầy cô muốn xếp gọn các hộp vào để còn đựng khi cần di chuyển. Biết rằng hộp A xếp lồng vào trong hộp B nếu kích thước 2 mặt đáy của A nhỏ hơn của B. Chiều cao của chúng không quan trọng.

**Yêu cầu:** Xác định số lượng nhiều nhất các hộp có thể lồng vào nhau.

**Dữ liệu vào:** File văn bản **XEPHCN.INP** có cấu trúc như sau:

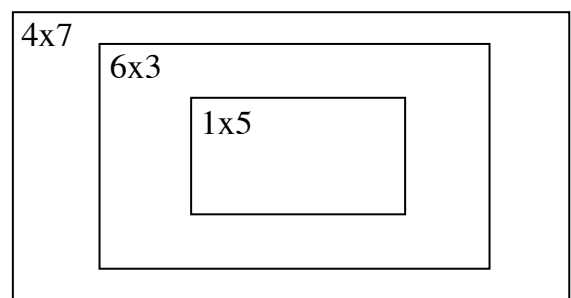
- Dòng 1: Ghi số nguyên dương N.
- Dòng thứ i trong N dòng tiếp theo ghi hai số nguyên dương  $a_i$   $b_i$ , là độ dài hai cạnh đáy của hộp thứ i.

**Dữ liệu ra:** File văn bản **XEPHCN.OUT**

Một số nguyên dương là số lượng lớn nhất các hộp có thể xếp lại với nhau.

**Ví dụ:**

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



**Giới hạn:**  $2 \leq N \leq 10^5$  trong đó 60% test có  $N \leq 10^3$

**Gợi ý:**

Thuật toán Quy Hoạch động

- Đổi các hình chữ nhật sao cho chiều rộng  $a \leq$  chiều dài  $b$
- Sắp xếp các hình sao cho chiều rộng tăng dần

- Tìm dãy con tăng dài nhất tạo thành từ chiều dài b. Số phần tử của dãy chính là số HCN nhiều nhất xếp được

#### Bài 4: Giấc mơ.

Sau một ngày làm việc vất vả, Miu con có một giấc ngủ thật ngon. Trong giấc mơ, Miu con mơ thấy mình đang tham gia một cuộc thi nơi mà vẻ đẹp trí tuệ được đánh giá cao hơn cả nhan sắc. Mỗi thí sinh tham gia thi phải trả lời n câu hỏi ở n vị trí được sắp xếp trên một hàng dọc được đánh số từ 1 đến n. Mỗi câu hỏi sẽ có  $A_i$  thời gian suy nghĩ. Mỗi thí sinh sẽ được chơi nhiều lượt cho đến khi trả lời đủ n câu hỏi. Tuy nhiên, ở mỗi lượt họ phải chọn K vị trí liên tiếp và trả lời hết các câu hỏi đó với thời gian trả lời là nhỏ nhất trong đó.

Ví dụ nếu dãy câu hỏi là 5 và có thời gian trả lời lần lượt là 4, 3, 4, 3, 4 độ dài mỗi đoạn trong mỗi lượt chơi là 3. Ở lượt đầu tiên nếu chọn đoạn [2, 4] thí sinh sẽ trả lời được hai câu có thời gian là 3. Lượt thứ hai sẽ chọn đoạn [1, 3] để trả lời 2 câu có thời gian là 4 và dùng lượt cuối cùng chọn đoạn [3, 5] để trả lời câu hỏi thứ 5. Ai dùng ít lượt chơi nhất sẽ là người chiến thắng.

**Yêu cầu:** Bạn hãy giúp Miu con sao cho số lượt chơi là ít nhất

**Dữ liệu vào: DREAM.INP**

- Dòng 1: Ghi 2 số nguyên dương N và K ( $1 \leq K \leq N \leq 100,000$ ).
- N dòng tiếp theo: Mỗi dòng ghi một số nguyên không âm  $A_i$  không quá 1,000,000,000 là thời gian trả lời câu hỏi thứ i.

**Dữ liệu ra: DREAM.OUT**

- Ghi ra kết quả tìm được.

**Ví dụ:**

DREAM.INP	DREAM.OUT
5 3	3
40	
30	
40	
30	
40	

#### Bài 5: Xe Buýt

Một xe buýt của công ty có nhiệm vụ đón nhân viên đến trụ sở làm việc. Trên hành trình xe buýt sẽ tiếp nhận nhân viên đứng chờ ở các điểm hẹn nếu như xe còn

chỗ trống. Xe buýt có thể sẽ đỗ lại để chờ những công nhân còn chưa kịp đến điểm hẹn. Cho biết thời điểm mà mỗi nhân viên đến điểm hẹn của mình và thời điểm qua mỗi điểm hẹn của xe buýt. Giải thiết rằng xe buýt đến điểm hẹn đầu tiên tại thời điểm 0, thời gian xếp khách lên xe coi như bằng 0. Hãy xác định khoảng thời gian ngắn nhất để xe buýt có thể chở một số lượng các nhân viên đến trụ sở làm việc lớn nhất có thể được.

**Dữ liệu** vào từ file **BUS.INP**:

- Dòng đầu tiên chứa hai số nguyên dương  $N$  và  $M$  theo thứ tự là số điểm hẹn và số chỗ ngồi của xe buýt.
- Dòng thứ  $i$  trong số  $N$  dòng tiếp theo chứa số nguyên  $t_i$  là thời gian cần thiết để xe buýt di chuyển từ điểm hẹn  $i$  đến điểm hẹn  $i+1$  (điểm hẹn thứ  $N+1$  sẽ là trụ sở làm việc của công ty), số nguyên  $K$  là số lượng nhân viên đến điểm hẹn  $i$  và tiếp đến là  $K$  số nguyên là các thời điểm đến điểm hẹn của  $K$  nhân viên.

Giới hạn:  $1 \leq M \leq 2000$ ,  $1 \leq N \leq 200000$

**Kết quả** ghi ra file văn bản **BUS.OUT** thời gian ngắn nhất tìm được.

*Ví dụ:*

BUS.INP	BUS.OUT
3 5 1 2 0 1 1 1 2 1 4 0 2 3 4	4
3 2 3 2 4 3 1 3 6 3 7 5 1 5	10

**Giải thích:** Trên đường đến công ty có 3 trạm xe buýt. Từ trạm 1 đến trạm 2, trạm 2 đến trạm 3, và từ trạm 3 đến công ty lần lượt mất 3, 1 và 5 đơn vị thời gian. Xe buýt có thể đi như sau: đến thẳng trạm 2, đón người thứ 2, đến trạm 3, chờ 1 đơn vị thời gian để đón người duy nhất ở trạm này, và cuối cùng đến công ty. Tổng cộng xe buýt đi mất  $3 + 1 + 1 + 5 = 10$  đơn vị thời gian.

**Gợi ý:**

- Đương nhiên khi xe buýt đi từ 1  $\rightarrow$  n+1 thì phải tốn khoảng thời gian để cho rời. Gọi biến res là nghiệm bài toán.
- Gọi  $F[i]$  là tổng thời gian xe bus đi từ trạm 1  $\rightarrow$  trạm i . Ban đầu  $F[1] = 0$  vì nó ở đây từ đầu rồi. Mỗi lần nhập một thời gian mới thì ta cứ cộng dồn biến thời gian vô thôi. Theo trên nói thì res có  $F[n+1]$
- Gọi  $a[i]$  là thời gian xe cần chờ người thứ i khi đến trạm. Tạo biến p là tổng số lượng người. Mỗi lần nhập thời gian một người đến tại một thời điểm thì ta tính  $a[i]$ . Tính bằng cách sau : dễ thấy nếu người đó đến trước khi xe đến thì  $a[i] = 0$  vì xe chẳng cần chờ. Còn nếu xe chờ thì nó sẽ phải chờ một khoảng là  $t[j] - f[i]$ . trong đó  $t[j]$  là thời gian người thứ j đến tại trạm thứ i và  $f[i]$  là thời gian xe đến trạm này.
- Bây giờ ta chỉ cần sort lại a thôi. Sau đó chọn ra thằng  $a[m]$  vì đây chính là thời gian mà xe cần phải chờ người thứ m sau khi sort hay nói cách khác là thời gian nhỏ nhất chờ m người

### Bài 1) Những vị khách sộp vào nhà hàng

Một nhà hàng bắt đầu mở cửa tại thời điểm 0 và đóng cửa tại thời điểm  $T=2.10^9$ . Tại cửa ra vào nhà hàng có treo một bảng hiện thị số. Tại thời điểm 0, số trên bảng là 0 và cứ sau 1s số trên bảng giữ nguyên giá trị hoặc tăng, giảm một đơn vị. Bảng chỉ hiện thị được các số không âm.

Có N vị khách sộp đi qua nhà hàng, vị khách sộp thứ i đi qua nhà hàng tại thời điểm  $T_i$  và sở thích của ông ta là số  $S_i$ . Nếu như ở thời điểm ông ta đi qua nhà hàng biến số trước cửa nhà hàng hiện đúng số ông ta thích thì ông ta sẽ vào và tiêu một số tiền là  $P_i$ .

Yêu cầu: Hãy giúp nhà hàng điều khiển bảng số sao cho tổng số tiền mà các vị khách sộp vào nhà hàng là lớn nhất.

Input: Cho trong file WELCOME.INP

- Dòng đầu tiên ghi số nguyên dương N ( $N \leq 200$ )
- Trong N dòng tiếp theo, dòng thứ i ghi thông tin về vị khách thứ i gồm ba số nguyên dương  $T_i$ ,  $S_i$  và  $P_i$ .

Output: Ghi ra file WELCOME.OUT một số nguyên duy nhất là tổng số tiền lớn nhất mà nhà hàng nhận được.

*Ví dụ:*

WELCOME . INP	WELCOME . OUT
3	7
2 1 3	

3 2 4	
1 3 10	