

## BÀI TẬP LẬP TRÌNH

### MỨC THƯỞNG

Cho danh sách gồm  $n$  công việc. Công việc thứ  $i$  có thời gian xử lý  $t_i$  và thời hạn hoàn thành  $d_i$  (deadline). Nếu bạn thực hiện xong công việc thứ  $i$  tại thời điểm  $f_i$  thì bạn sẽ nhận được mức thưởng  $d_i - f_i$  (mức thưởng có thể bị âm, khi đó được xem là mức phạt). Thời điểm được tính bắt đầu từ 0 và bạn phải hoàn thành tất cả  $n$  công việc theo một thứ tự nào đó một cách tuần tự.

**Yêu cầu:** Hãy tính tổng mức thưởng tối đa có thể nhận được.

**Dữ liệu:** Vào từ tập tin văn bản **REWARD.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) – số công việc được giao.
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa 2 số nguyên  $t_i, d_i$  ( $1 \leq t_i, d_i \leq 10^6$ ) – mô tả một công việc.

**Kết quả:** Ghi ra tập tin văn bản **REWARD.OUT** tổng mức thưởng tối đa có thể nhận được.

**Ví dụ:**

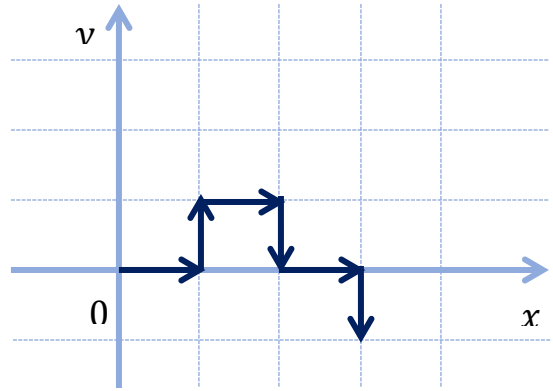
REWARD . INP	REWARD . OUT
3 6 10 8 15 5 12	2

## ROBOT DI CHUYỂN

Cho lưới nguyên Oxy. Điểm nguyên  $(x_1, y_1)$  và điểm nguyên  $(x_2, y_2)$  được gọi là kề nhau nếu thỏa điều kiện  $|x_1 - x_2| + |y_1 - y_2| = 1$ .

Một robot ban đầu đứng tại gốc tọa độ. Ở mỗi bước, robot sẽ di chuyển sang một điểm nguyên kề với vị trí hiện tại.

Từ bước di chuyển thứ 2 trở đi, robot có thể đi tiếp theo hướng cũ, rẽ sang trái, rẽ sang phải hay trở lại vị trí trước đó.



Trong ví dụ ở hình bên, từ ô  $(0, 0)$ , robot đi đến  $(1, 0)$ , rẽ trái sang ô  $(1, 1)$ , rẽ phải sang ô  $(2, 1)$ , rẽ phải sang ô  $(2, 0)$ , rẽ trái sang ô  $(3, 0)$  cuối cùng rẽ phải sang ô  $(3, -1)$ .

**Yêu cầu:** cho tọa độ các điểm nguyên mà robot đã đi qua. Hãy đếm xem robot đã rẽ phải bao nhiêu lần.

**Dữ liệu:** Vào từ tập tin văn bản **ROBOT.INP** có cấu trúc sau:

- Dòng đầu tiên chứa 1 số nguyên dương  $n (2 \leq n \leq 10^5)$  là tổng số điểm nguyên mà robot đã đi qua (kể cả vị trí xuất phát là gốc tọa độ)
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa 2 số nguyên  $x_i, y_i$  là tọa độ điểm nguyên mà robot đã đi qua.

**Kết quả:** Ghi ra tập tin văn bản **ROBOT.OUT** gồm một số nguyên là số lần robot đã rẽ phải

**Ví dụ:**

ROBOT . INP	ROBOT . OUT
7 0 0 1 0 1 1 2 1 2 0 3 0 3 -1	3

**LẬP LỊCH**

Bạn cần thực thi càng nhanh càng tốt  $n$  chương trình máy tính đánh số  $1, 2, \dots, n$ . Chương trình thứ  $i$  mất  $s_i$  đơn vị thời gian cài đặt và  $r_i$  đơn vị thời gian thực thi. Chiếc máy tính của bạn được trang bị bộ vi xử lý mạnh mẽ nhất hiện nay cho phép thực hiện nhiều công việc cũng như chạy nhiều chương trình cùng một lúc mà không hề chậm đi. Tuy nhiên, việc cài đặt một chương trình đòi hỏi sự tập trung chú ý cao nên bạn không thể tiến hành cài đặt hai chương trình trong cùng thời điểm. Khi một chương trình được cài xong, nó sẽ tự động chạy ngay lập tức và bạn có thể chuyển sang cài chương trình khác, thời gian chuyển coi như không đáng kể.

Hãy xác định xem bạn có thể chạy xong tất cả  $n$  chương trình trong thời gian ngắn nhất là bao nhiêu.

**Dữ liệu:** Vào từ tập tin văn bản **SCHEDULE.INP**

- Dòng đầu ghi  $n (1 \leq n \leq 10^5)$ .
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo ghi hai số nguyên  $s_i, r_i (1 \leq s_i, r_i \leq 10^9)$ .

**Kết quả:** Ghi ra tập tin văn bản **SCHEDULE.OUT** tổng thời gian ngắn nhất tìm được.

**Ví dụ:**

SCHEDULE . INP	SCHEDULE . OUT
4 4 2 2 2 1 2 3 4	12

*Giải thích:* thực thi theo thứ tự (3, 2, 4, 1).

## LŨY THỪA ĐÚNG

Số nguyên  $x$  được gọi là số lũy thừa đúng nếu tồn tại hai số nguyên  $a, b (b > 1)$  sao cho  $x = a^b$ . Ví dụ 16 là một số lũy thừa đúng vì  $16 = 2^4$ .

Việc biểu diễn  $x$  dưới dạng lũy thừa có thể không duy nhất. Ví dụ  $16 = 2^4 = 4^2$ . Như vậy, với  $x = 16$  tồn tại 2 cặp số  $(a, b)$  biểu diễn nó.

**Yêu cầu:** Cho số nguyên  $x (1 \leq x \leq 10^{18})$ . Hãy tìm tất cả cặp số nguyên  $(a, b)$  thỏa  $x = a^b$ .

**Dữ liệu:** Vào từ tập tin văn bản **POWER.INP** chứa số nguyên  $x$ .

**Kết quả:** Ghi ra tập tin văn bản **POWER.OUT**

- Dòng đầu tiên là số nguyên  $k$  là số cặp tìm được.
- $k$  dòng tiếp theo liệt kê các cặp  $(a, b)$  thỏa yêu cầu có thứ tự tăng dần theo  $b$ .

**Ví dụ:**

POWER . INP	POWER . OUT
16	2 4 2 2 4

**CHUỖI ANAGRAM**

Hai chuỗi kí tự được gọi là anagram nếu ta có thể thay đổi thứ tự các kí tự trong chuỗi này để tạo thành chuỗi kia và ngược lại. Chẳng hạn các cặp chuỗi sau là anagram: ("taste", "state"), ("act", "cat"), ("listen", "slient"). Còn cặp chuỗi sau không phải là anagram: ("feel", "fell").

Xét chuỗi  $s$ , ta chia  $s$  thành 2 chuỗi con liên tiếp  $x$  và  $y$ , nghĩa là  $x = s_0s_1 \dots s_{i-1}$ ,  $y = s_is_{i+1} \dots s_{|s|-1}$  với  $i(0 < i < |s|)$  là vị trí tách. Nếu  $x, y$  không phải anagram ta sẽ thay thế một số kí tự để chúng trở thành anagram.

Chẳng hạn  $s = \text{"abccde"}$ , ta tách  $s$  thành  $x = \text{"abc"}$  và  $y = \text{"cde"}$ . Nếu thay kí tự 'a' bằng kí tự 'd', thay kí tự 'b' bằng kí tự 'e' trong chuỗi  $x$  thì  $x, y$  trở thành anagram với 2 phép thay thế.

**Yêu cầu:** Cho chuỗi  $s$ . Tìm số phép thay thế ít nhất để  $x, y$  trở thành anagram.

**Dữ liệu:** Vào từ tập tin văn bản **ANAGRAM.INP**

- Dòng đầu tiên chứa số nguyên  $t(1 \leq t \leq 100)$  – số bộ test.
- Mỗi dòng trong  $t$  dòng tiếp theo chuỗi  $s$  độ dài không quá 10000 các kí tự chữ cái latin in thường.

**Kết quả:** Ghi ra tập tin văn bản **ANAGRAM.OUT** gồm  $t$  dòng, mỗi dòng ghi số phép thay thế ít nhất cần thực hiện. Trường hợp không có lời giải thì ghi số  $-1$ .

**Ví dụ:**

ANAGRAM . INP	ANAGRAM . OUT
6	3
aaabbb	1
ab	-1
abc	2
mno	0
xyyx	1
xaxbbbxx	

## TẠO CHUỖI ANAGRAM

Hai chuỗi kí tự được gọi là anagram nếu ta có thể thay đổi thứ tự các kí tự trong chuỗi này để tạo thành chuỗi kia và ngược lại. Chẳng hạn các cặp chuỗi sau là anagram: ("taste", "state"), ("act", "cat"), ("listen", "slient"). Còn cặp chuỗi sau không phải là anagram: ("feel", "fell").

Còn cặp chuỗi sau không phải là anagram: ("computer", "poetry"). Tuy nhiên ta có thể xóa các kí tự 'c', 'm', 'u', ở chuỗi thứ nhất và kí tự 'y' ở chuỗi thứ hai để chúng trở thành cặp anagram ("opter", "poetr"). Như vậy ta cần 4 phép xóa.

**Yêu cầu:** Cho 2 chuỗi  $s_1$  và  $s_2$ . Tìm số kí tự ít nhất cần phải xóa để 2 chuỗi trở thành anagram.

**Dữ liệu:** Vào từ tập tin văn bản **ANADEL.INP**

- Dòng đầu tiên chứa chuỗi  $s_1$ .
- Dòng thứ hai chứa chuỗi  $s_2$ .
- Cả 2 chuỗi có độ dài không quá  $10^4$  các kí tự chữ cái latin in thường.

**Kết quả:** Ghi ra tập tin văn bản **ANADEL.OUT** một số nguyên là số kí tự ít nhất cần xóa.

**Ví dụ:**

ANADEL . INP	ANADEL . OUT
computer poetry	4

## KIỂM TRA PALINDROME

Chuỗi palindrome là chuỗi có giá trị giống nhau khi đọc từ trái sang hoặc từ phải sang, chẳng hạn chuỗi "abba". Một số chuỗi có thể biến thành palindrome nếu thay đổi thứ tự các kí tự trong chuỗi, một số chuỗi khác thì không. Chẳng hạn chuỗi "aabbcc" có thể tạo thành chuỗi palindrome "abccba", còn chuỗi "aaabcc" thì không thể.

**Yêu cầu:** Cho chuỗi  $s$  có độ dài không quá  $10^5$  kí tự chữ cái latin in thường. Hãy cho biết có thể thay đổi thứ tự các kí tự trong  $s$  để trở thành chuỗi palindrome.

**Dữ liệu:** Vào từ tập tin văn bản **CHECKPAL.INP** chứa chuỗi  $s$ .

**Kết quả:** Ghi ra tập tin văn bản **CHECKPAL.OUT** thông báo YES hoặc NO tương ứng với câu trả lời.

**Ví dụ:**

CHECKPAL . INP	CHECKPAL . OUT
aabbcc	YES
aaabcc	NO

## CHUỖI HỢP LỆ

Một chuỗi được gọi là hợp lệ nếu tất cả kí tự trong chuỗi có tần số xuất hiện bằng nhau. Một chuỗi nếu có thể xóa đúng 1 kí tự tại đúng 1 vị trí nào đó để chuỗi trở thành hợp lệ thì cũng được gọi là chuỗi hợp lệ.

Chẳng hạn chuỗi  $s = \text{"abc"}$  là chuỗi hợp lệ, chuỗi  $s = \text{"acbc"}$  cũng là chuỗi hợp lệ vì ta có thể xóa 1 kí tự c để trở thành  $s = \text{"abc"}$  hoặc  $s = \text{"acb"}$  đều là chuỗi hợp lệ. Chuỗi  $s = \text{"abccc"}$  không phải là chuỗi hợp lệ.

**Yêu cầu:** Cho chuỗi  $s$  độ dài không quá  $10^5$  kí tự latin in thường. Cho biết  $s$  có phải là chuỗi hợp lệ hay không.

**Dữ liệu:** Vào từ tập tin văn bản **VALIDSTR.INP** chứa chuỗi  $s$ .

**Kết quả:** Ghi ra tập tin văn bản **VALIDSTR.OUT** thông báo YES hoặc NO cho biết câu trả lời tương ứng.

**Ví dụ:**

VALIDSTR.INP	VALIDSTR.OUT
abcdefghijklghgfedecba	YES
aabbcd	NO



## ĐOẠN CON LỚN NHẤT

Cho dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$ . Hãy tìm tổng lớn nhất của một đoạn con gồm các phần tử liên tiếp của dãy.

**Dữ liệu:** Vào từ tập tin văn bản **MAXSUBARRAY.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) – số phần tử của dãy.
- Dòng tiếp theo chứa dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **MAXSUBARRAY.OUT** một số nguyên là tổng lớn nhất tìm được.

**Ví dụ:**

MAXSUBARRAY . INP	MAXSUBARRAY . OUT
8 -1 3 -2 5 3 -5 2 2	9

## LIÊN HOAN PHIM

Liên hoan phim năm nay có  $n$  phim được vào diện xét chọn. Có 2 giải chính là giải đạo diễn hay nhất và giải phim có nội dung hay nhất, mỗi giải được trao cho một phim. Hai giải này được trao cho 2 phim khác nhau.

Kết quả thăm dò ý kiến khán giả và các nhà bình luận điện ảnh xác định được mức độ hân hoan của công chúng khi một phim được giải này hay giải khác hoặc không được giải nào, trong đó  $a_i$  là mức độ hân hoan khi phim thứ  $i$  không được giải nào,  $b_i$  là mức độ hân hoan khi phim được trao giải đạo diễn hay nhất và  $c_i$  là mức độ hân hoan khi phim được giải nội dung hay nhất.

**Yêu cầu:** Hãy xác định tổng độ hân hoan lớn nhất mà cuộc bầu chọn có thể mang lại cho công chúng và chỉ ra phim nào sẽ được giải gì trong trường hợp này.

**Dữ liệu:** Vào từ tập tin văn bản **CINEMA.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $2 \leq n \leq 10^5$ ).
- Dòng thứ  $i$  trong  $n$  dòng sau chứa 3 số nguyên  $a_i, b_i, c_i$  ( $1 \leq a_i, b_i, c_i \leq 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **CINEMA.OUT**

- Dòng thứ nhất chứa một số nguyên – tổng độ hân hoan lớn nhất có thể đạt được.
- Dòng thứ 2 chứa 2 số nguyên xác định phim được giải đạo diễn hay nhất và phim được giải nội dung hay nhất. Nếu tồn tại nhiều cặp chỉ số cùng thỏa mãn thì đưa ra cặp có thứ tự từ điển nhỏ nhất.

**Ví dụ:**

CINEMA . INP	CINEMA . OUT
3	17
3 6 9	2 3
1 5 7	
1 3 9	

## THỰC KHÁCH NHÀ HÀNG

Có  $n$  thực khách đến thưởng thức các món đặc sản của một nhà hàng. Với mỗi thực khách, ta biết thời điểm vào và thời điểm rời nhà hàng. Hãy cho biết số lượng nhiều nhất các thực khách có mặt trong nhà hàng tại một thời điểm nào đó.

**Dữ liệu:** Vào từ tập tin văn bản **RESTAURANT.INP**

- Dòng đầu tiên chứa số nguyên  $n (1 \leq n \leq 2 \times 10^5)$  – số lượng thực khách.
- Mỗi dòng trong  $n$  dòng tiếp theo chứa 2 số nguyên  $a, b (1 \leq a < b \leq 10^9)$  – thời điểm vào và thời điểm ra khỏi nhà hàng của thực khách tương ứng.

**Kết quả:** Ghi ra tập tin văn bản **RESTAURANT.OUT** một số nguyên là số thực khách nhiều có mặt trong nhà hàng ở một thời điểm nào đó.

**Ví dụ:**

RESTAURANT . INP	RESTAURANT . OUT
3 5 8 2 4 3 9	2

**PYTHAGORE**

Định lí Pythagore do Pythagoras, nhà toán học người Hy Lạp sống vào khoảng thế kỉ thứ 6 trước công nguyên, phát minh. Ngày nay, định lí Pythagore được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như kiến trúc, xây dựng và đo lường. Những ứng dụng phổ biến của định lí Pythagore có thể kể ra là tính khoảng cách giữa 2 điểm trong mặt phẳng hoặc kiểm tra một tam giác có phải là tam giác vuông.

Định lí phát biểu như sau: Tam giác vuông có độ dài 3 cạnh  $a, b, c$ , trong đó  $c$  là độ dài cạnh huyền khi đó  $a^2 + b^2 = c^2$ .

Tí rất thích thú với tính chất này nhưng cậu ta chỉ quan tâm đến các tam giác vuông có độ dài ba cạnh  $a, b, c$  ( $0 < a < b < c$ ) là số nguyên. Tí nhận thấy rằng, với cùng một chu vi thì có thể tồn tại nhiều tam giác vuông khác nhau có cạnh nguyên. Chẳng hạn với chu vi 60, tồn tại 2 tam giác vuông là (10,24,26) và (15,20,25). Còn với chu vi 12 chỉ tồn tại 1 tam giác vuông (3,4,5).

**Yêu cầu:** Cho trước chu vi  $P$ . Đếm số lượng tam giác vuông khác nhau có độ dài ba cạnh nguyên  $a, b, c$  ( $a < b < c$ ) và chu vi là  $P$ .

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

- Dòng đầu tiên chứa số nguyên  $t$  ( $1 \leq t \leq 5000$ ) – số bộ test.
- Dòng thứ  $i$  trong  $t$  dòng tiếp theo chứa số nguyên  $P_i$  ( $1 \leq P_i \leq 5000$ ) – chu vi của các tam giác vuông thứ  $i$ .

**Kết quả:** Ghi ra file văn bản **PYTHAGORE.OUT** gồm  $t$  dòng, dòng thứ  $i$  ghi một số nguyên là số lượng tam giác vuông khác nhau có độ dài ba cạnh nguyên và chu vi  $P_i$ .

**Ví dụ:**

PYTHAGORE . INP	PYTHAGORE . OUT
2	1
12	2
60	

## TÍCH LỚN NHẤT

Cho dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$ . Hãy tìm 2 vị trí  $i, j (i \neq j)$  sao cho  $a_i \times a_j$  đạt giá trị lớn nhất nhưng không được vượt quá giá trị  $T$  cho trước.

**Dữ liệu:** Vào từ tập tin văn bản **MAXPRODUCT.INP**

- Dòng đầu chứa 2 số nguyên  $n, T (1 \leq n \leq 2 \times 10^5; 1 \leq T \leq 10^{18})$ .
- Dòng thứ hai chứa dãy  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^{18})$ .

**Kết quả:** Ghi ra tập tin văn bản **MAXPRODUCT.OUT**

- Dòng đầu tiên ghi giá trị tích lớn nhất thỏa yêu cầu tìm được. Nếu không tìm được cặp số thỏa mãn yêu cầu thì ghi số 0.
- Dòng tiếp theo ghi 2 số nguyên  $i$  và  $j$  là vị trí của 2 số nguyên tìm được. Nếu tồn tại nhiều kết quả thì ghi kết quả bất kỳ.

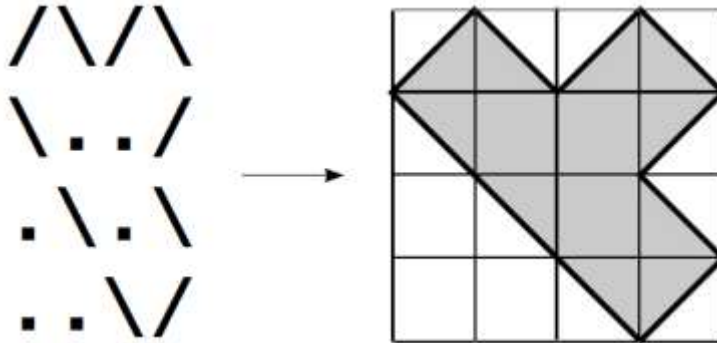
**Ví dụ:**

MAXPRODUCT . INP	MAXPRODUCT . OUT
4 16 8 1 4 1	8 1 2
4 16 16 8 4 2	16 2 4

**ĐA GIÁC ASCII**

Một bức tranh vẽ một đa giác bằng 3 kí tự ASCII ".", "/", và "\". Trong đó kí tự "." biểu diễn một ô vuông đơn vị. Kí tự "/" biểu diễn một đường chéo từ góc phải trên đến góc trái dưới của hình vuông đơn vị. Kí tự "\" biểu diễn một đường chéo từ góc trái trên đến góc phải dưới của hình vuông đơn vị.

Chẳng hạn hình đa giác minh họa bên dưới được biểu diễn bởi các kí tự ASCII.



**Yêu cầu:** Hãy tính diện tích của đa giác được mô tả bởi 3 kí tự như trên.

**Dữ liệu:** Vào từ tập tin văn bản **ASCII.INP**

- Dòng đầu tiên chứa 2 số nguyên  $h, w$  ( $2 \leq h, w \leq 100$ ) – chiều cao và ngang của bức tranh.
- Mỗi dòng trong  $h$  dòng chứa  $w$  kí tự ASCII ".", "/", và "\".
- Dữ liệu đảm bảo bức tranh chỉ chứa 1 đa giác không tự cắt.

**Kết quả:** Ghi ra tập tin văn bản **ASCII.OUT** một số nguyên là diện tích của đa giác.

**Ví dụ:**

ASCII.INP	ASCII.OUT
<pre> 4 4 /\ /\ \.. / .\. \ ..\ \/ </pre>	8

**CỬA MÁY**

Một hàng cây gồm  $n$  cây đánh số từ 1 tới  $n$ , cây thứ  $i$  có chiều cao  $h_i$ . Người ta muốn khai thác gỗ từ những cây này bằng một máy cưa. Máy cưa vận hành như sau: Trước hết phải thiết lập một độ cao  $\Delta$  cho lưỡi cưa, sau đó di chuyển máy cưa qua hàng cây. Mỗi khi máy cưa đi qua cây độ cao  $h > \Delta$  thì cây đó bị cưa còn lại chiều cao  $\Delta$  và người ta lấy được  $h - \Delta$  mét gỗ từ cây này. Dĩ nhiên những cây có độ cao  $\leq \Delta$  không bị cưa và người ta không lấy được gỗ từ những cây đó.

**Yêu cầu:** Cho dãy số nguyên dương  $m_1, m_2, \dots, m_k$ . Với mỗi giá trị  $m_j$ , tìm số nguyên  $\Delta_j$  lớn nhất sao cho nếu đặt độ cao của lưỡi cưa là  $\Delta_j$  thì tổng số mét gỗ khai thác được không ít hơn  $m_j$  ( $j = 1, 2, \dots, k$ ).

**Dữ liệu:** Vào từ tập tin văn bản **SAW.INP**

- Dòng đầu tiên chứa hai số nguyên dương  $n, k \leq 10^5$ .
- Dòng thứ hai chứa  $n$  số nguyên dương  $h_1, h_2, \dots, h_n$  ( $\forall i: h_i \leq 10^6$ ).
- Dòng thứ ba chứa  $k$  số nguyên dương  $m_1, m_2, \dots, m_k$  ( $\forall j: m_j \leq \sum_{i=1}^n h_i$ ).

**Kết quả:** Ghi ra tập tin văn bản **SAW.OUT** một dòng  $k$  số nguyên  $\Delta_1, \Delta_2, \dots, \Delta_k$  tìm được.

**Ví dụ:**

SAW . INP	SAW . OUT
4 2 20 15 10 17 7 4	15 16

**Giải thích:**

- Nếu đặt độ cao lưỡi cưa là 15, ta khai thác được 7m gỗ: 5 mét từ cây 1 và 2 mét từ cây 4.
- Nếu đặt độ cao lưỡi cưa là 16, ta khai thác được 5m gỗ: 4 mét từ cây 1 và 1 mét từ cây 4.

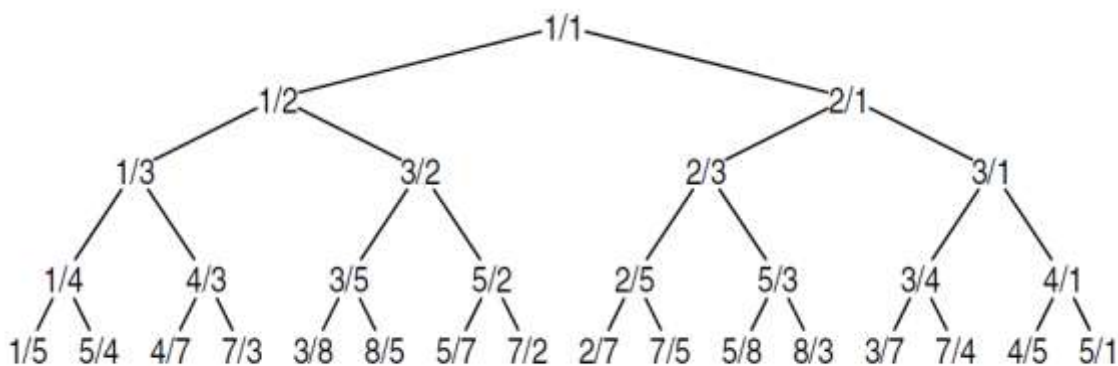
## CÂY VÀ PHÂN SỐ

Bờm rất thích nghiên cứu về cây nhị phân và phân số. Bờm xây dựng một cây nhị phân có các nút được đánh số thứ tự như sau:

- Nút gốc có thứ tự 1.
- Nút thứ  $i$  có 2 nút con: nút con trái có thứ tự  $2i$  và nút con phải có thứ tự  $2i + 1$ .

Trên cây nhị phân này, Bờm đặt các giá trị vào các nút như sau: (như hình vẽ)

- Nút gốc 1 mang giá trị là phân số  $\frac{1}{1}$ .
- Nếu nút  $i$  mang giá trị là phân số  $\frac{r}{s}$  thì nút con trái của nó mang giá trị  $\frac{r}{r+s}$ , nút con phải mang giá trị  $\frac{r+s}{s}$ .



**Yêu cầu:** Cho số nguyên dương  $n$ . Hãy cho biết nút thứ  $n + 1$  mang giá trị là phân số nào?

**Dữ liệu:** Vào từ tập tin văn bản **FRACTION.INP** chứa số nguyên  $n$  ( $0 \leq n \leq 2 \times 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **FRACTION.OUT** 2 số nguyên  $r, s$  là giá trị của nút  $n + 1$ . Các số nguyên ghi cách nhau 1 dấu cách.

**Ví dụ:**

FRACTION . INP	FRACTION . OUT
0	1 1
5	2 3
10	5 2



**CHƠI ĐÀN GHI TA**

Trong cuộc thi sáng tạo robot, Steve đã trình diễn robot chơi ghi-ta của mình, một robot, theo lời mô tả của tác giả “có đến hàng tỷ ngón tay”.

Đàn ghi ta có 6 dây đánh số từ 1 đến 6 và có  $p$  phím đánh số từ 1 đến  $p$ . Khi chơi nhạc người ta gảy dây và bấm phím để có các giai điệu khác nhau. Với mỗi dây, nếu có nhiều phím cùng được bấm thì âm điệu sẽ được quyết định bởi phím có số cao nhất. Ví dụ, âm điệu khi gảy dây số 3 và bấm đồng thời các phím 5 và 7 sẽ giống như khi ta gảy dây này và chỉ bấm phím số 7.

Điều khó khăn nhất khi biểu diễn một bài nhạc là điều khiển cường độ tức phải gảy dây nào và bấm phím nào, còn điều khiển trường độ (thời gian phát một nốt) không phải là quá khó về phương diện kỹ thuật.

Chương trình điều khiển cường độ của Steve được tối ưu hóa theo hướng giảm số lần chuyển tay bấm phím xuống còn ít nhất.

Xét bản nhạc có  $n$  nốt, nốt thứ  $i$  được cho bởi cặp giá trị nguyên  $(s_i, f_i)$ , trong đó  $s_i$  – dây cần gảy và  $f_i$  – phím cần bấm ( $1 \leq s_i \leq 6$ ;  $1 \leq f_i \leq p$ ). Bản nhạc cần chơi theo trình tự thực hiện lần lượt từ nốt 1 đến nốt  $n$ . Ví dụ, với  $n = 5$ ,  $p = 15$  và bản nhạc là  $(2,8), (2,10), (2,12), (2,10), (2,5)$ , thao tác bấm phím của robot là bấm phím 8, bấm phím 10, bấm phím 12, nhả phím 12, nhả phím 10, nhả phím 8 và bấm phím 5, tất cả là 7 thao tác, thực hiện trên dây số 2.

**Yêu cầu:** Cho  $n, p$  và các  $s_i, f_i$  ( $1 \leq n \leq 5 \times 10^5$ ;  $2 \leq p \leq 3 \times 10^5$ ). Hãy xác định số thao tác bấm phím ít nhất cần thực hiện.

**Dữ liệu:** Vào từ tập tin văn bản **GUITAR.INP**

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $p$ ,
- Dòng thứ  $i$  trong  $n$  dòng sau chứa hai số nguyên  $s_i$  và  $f_i$ .

**Kết quả:** Ghi ra tập tin văn bản **GUITAR.OUT** một số nguyên – số thao tác bấm phím ít nhất cần thực hiện.

**Ví dụ:**

GUITAR . INP	GUITAR . OUT
5 15 2 8 2 10 2 12 2 10 2 5	7

**THU DỌN SỎI**

Cho  $n(1 \leq n \leq 10^6)$  đồng sỏi xếp thành một dãy hàng ngang, đồng thứ  $i$  có  $a_i$  viên sỏi ( $1 \leq a_i \leq 10^9$ ). Ta thực hiện thu dọn  $n$  đồng sỏi về còn 1 đồng theo nguyên tắc sau: xét 2 đồng sỏi liên tiếp nhau, đồng có số lượng sỏi nhiều hơn sẽ được giữ lại và đồng còn lại có số lượng sỏi ít hơn sẽ bị loại bỏ đi. Chi phí để thực hiện thao tác này bằng số lượng sỏi trong đồng nhiều hơn được giữ lại. Như vậy, sau  $n - 1$  lần thu dọn thì chỉ còn lại 1 đồng sỏi. Chi phí thu dọn sỏi được tính bằng tổng chi phí của tất cả  $n - 1$  lần thực hiện.

**Yêu cầu:** Hãy xác định chi phí nhỏ nhất để thu dọn  $n$  đồng sỏi về còn 1 đồng sỏi.

**Dữ liệu:** Vào từ tập tin văn bản **MINCOST.INP**

- Dòng đầu tiên chứa số nguyên  $n$ ,
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa số nguyên  $a_i$ .

**Kết quả:** Ghi ra tập tin văn bản **MINCOST.OUT** chi phí nhỏ nhất tìm được.

**Ví dụ:**

MINCOST . INP	MINCOST . OUT
3 1 2 3	5

**GIẢI THƯỞNG**

Jane và Steve đều giành giải nhất trong một cuộc thi về học thuật. Ban tổ chức muốn thách thức cả hai thêm một lần nữa nên không trao thưởng trực tiếp mà đưa ra một dãy  $n$  phần thưởng có trị giá tương ứng  $a_1, a_2, \dots, a_n$ . Vì là nữ nên Jane được ban tổ chức cho phép chọn trước và lấy đi  $k$  phần thưởng nằm liên tiếp, sau đó Steve sẽ lấy  $k$  phần thưởng liên tiếp trong số các phần thưởng nằm liên tiếp còn lại.

Vì không ưa Steve cho lắm nên Jane tìm cách chọn trước những phần thưởng sao cho tổng trị giá phần thưởng lớn nhất của Steve nhận được là nhỏ nhất, bất chấp phần thưởng Jane nhận được là gì.

**Yêu cầu:** Hãy xác định giá trị  $x$  nhỏ nhất sao tổng giá trị phần thưởng lớn nhất của Steve không vượt quá  $x$ .

**Dữ liệu:** Vào từ tập tin văn bản **PRIZES.INP**

- Dòng đầu tiên chứa số hai nguyên  $n, k$  ( $3 \leq n \leq 10^5$ ;  $1 \leq k \leq n/3$ )
- Dòng tiếp theo chứa dãy  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )

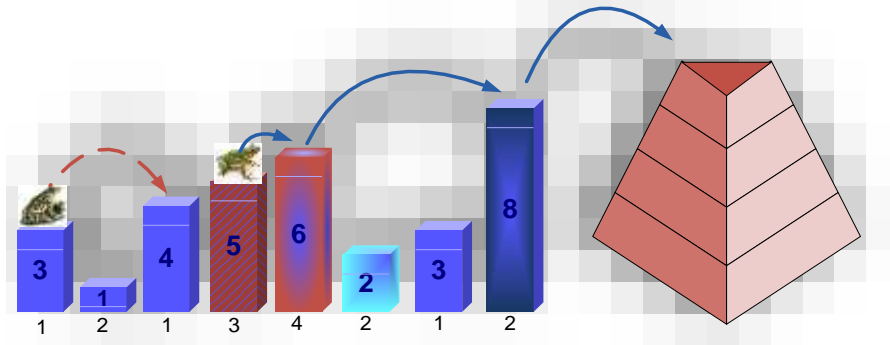
**Kết quả:** Ghi tập tin văn bản **PRIZES.OUT** số nguyên  $x$ .

**Ví dụ:**

PRIZES . INP	PRIZES . OUT
10 2 1 2 4 5 2 4 2 2 1 6	7

## ẾCH ĐỘT BIẾN GEN

Cuộc sống an nhàn với thức ăn đầy đủ và đa dạng tại các đồng rác thành phố đã sinh ra thể hệ các chú ếch đột biến gen. Trên con đường dẫn đến bãi rác thành phố có  $n(n \leq 10^6)$



đồng rác, đánh số bắt đầu từ 0 đến  $n-1$  từ trái qua phải. Đồng rác thứ  $i$  có độ cao  $h_i (0 \leq i \leq n-1; 0 < h_i \leq 10^9)$ . Trên mỗi đồng rác hiện có một chú ếch sống. Đến tuổi trưởng thành, mỗi chú ếch đều muốn đi tìm một chỗ sống tốt đẹp hơn bằng cách nhảy sang đồng rác cao hơn gần nhất bên phải. Chú ếch ở đồng rác thứ  $i$  có thể thực hiện được  $j_i$  bước nhảy ( $0 < j_i < n$ ). Bãi rác thành phố có độ cao lớn hơn mọi đồng rác trên đường. Ta ký hiệu độ cao này là -1 (vì không cần và cũng không thể biết chính xác).

Ví dụ, có 8 đồng rác với độ cao tương ứng từ trái sang phải là 3, 1, 4, 5, 6, 2, 3 và 8. Số bước nhảy mỗi chú ếch có thể thực hiện là 1, 2, 1, 3, 4, 2, 1, 2. Sau khi di chuyển hết khả năng của mình, chú ếch ở đồng rác 0 sẽ tới được đồng rác 2 với độ cao là 4, còn chú ếch ở đồng rác 3 – tới được bãi rác thành phố (độ cao -1).

**Yêu cầu:** Hãy xác định độ cao nơi ở mới của mỗi chú ếch.

**Dữ liệu:** Vào từ tập tin văn bản **FROGS.INP**

- Dòng đầu tiên chứa số nguyên  $n$
- Dòng thứ 2 chứa  $n$  số nguyên  $h_0, h_1, \dots, h_{n-1}$
- Dòng thứ 3 chứa  $n$  số nguyên  $j_0, j_1, \dots, j_{n-1}$

**Kết quả:** Ghi ra tập tin văn bản **FROGS.OUT** dãy gồm  $n$  số nguyên là độ cao nơi ở mới của mỗi chú ếch.

**Ví dụ:**

FROGS . INP	FROGS . OUT
8	4 5 5 -1 -1 8 8 -1
3 1 4 5 6 2 3 8	
1 2 1 3 4 2 1 2	

## HOÁN VỊ TỐI THIỂU

Cho dãy số nguyên  $a_1, a_2, \dots, a_n$  là một hoán vị của các số  $\{1, 2, \dots, n\}$ . Ta được phép thực hiện các phép hoán vị để đổi chỗ 2 phần tử bất kỳ cho nhau.

**Yêu cầu:** Hãy cho biết số phép hoán vị ít nhất để biến đổi dãy có thứ tự tăng dần.

**Dữ liệu:** Vào từ tập tin văn bản **MINSWAP.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 10^5$ ).
- Dòng thứ hai chứa dãy số nguyên phân biệt  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ )

**Kết quả:** Ghi ra tập tin văn bản **MINSWAP.OUT** một số nguyên là số phép hoán vị ít nhất cần thực hiện.

**Ví dụ:**

MINSWAP . INP	MINSWAP . OUT	Giải thích
7 1 3 2 4 5 6	5	1 4 1 2 4 5 5 6 6 7

## CHUỖI LẶP

Xét chuỗi ban đầu  $s$ . Gọi  $T(s, n)$  là chuỗi gồm  $n$  kí tự đầu tiên của chuỗi được tạo thành bằng cách viết lặp lại  $s$  vô hạn lần.

Chẳng hạn với  $s = \text{"abcac"}$  và với  $n = 10$  ta có  $T(s, n) = \text{"abcacabcac"}$ . Kí tự 'a' xuất hiện 4 lần trong chuỗi  $T$ .

**Yêu cầu:** Cho chuỗi  $s$  chỉ gồm các kí tự chữ cái latin in thường và số nguyên dương  $n$ . Hãy đếm tần số xuất hiện của kí tự 'a' trong chuỗi  $T(s, n)$ .

**Dữ liệu:** Vào từ tập tin văn bản **REPSTR.INP**

- Dòng đầu tiên chứa chuỗi  $s$  có độ dài không quá 100.
- Dòng thứ hai chứa số nguyên dương  $n (1 \leq n \leq 10^{12})$ .

**Kết quả:** Ghi ra tập tin văn bản **REPSTR.OUT** một số nguyên là kết quả của bài toán.

**Ví dụ:**

REPSTR . INP	REPSTR . OUT
aba 10	7
a 1000000	1000000

**XẾP HẠNG**

Tại buổi tựu trường đầu năm, một lớp gồm  $n$  học sinh đứng xếp thành một hàng dọc quay mặt về phía trước để nghe thông báo. Các học sinh đứng sau có thể nhìn thấy được những người đứng trước mình, ngược lại thì không, người đứng đầu hàng đương nhiên không thể thấy được những người xếp hàng sau mình.

Các học sinh thường thích so kè nhau về chiều cao nên hay nhìn xem những ai cao hơn mình để sắp thứ hạng chiều cao trong hàng. Tuy nhiên, điều này hơi khó khăn vì mỗi người chỉ nhìn thấy được số người đứng đằng trước mà cao hơn mình chứ không biết được những người đứng đằng sau. Bạn hãy giúp từng học sinh biết được mình cao thứ hạng mấy trong hàng. Không có 2 học sinh nào có cùng chiều cao.

**Yêu cầu:** Cho dãy  $a_i$  ứng với số người cao hơn mình mà học sinh đứng ở vị trí thứ  $i$  ( $2 \leq i \leq n$ ) nhìn thấy được, hãy cho biết chiều cao của học sinh đứng ở vị trí  $i$  xếp thứ hạng mấy trong hàng.

**Dữ liệu:** Vào từ tập tin văn bản **HRANK.INP**

- Dòng đầu tiên chứa số nguyên  $n, q$  ( $n \leq 10^3$ ) – số lượng học sinh và số lượng truy vấn.
- Dòng thứ hai chứa dãy gồm  $n - 1$  số nguyên:  $a_2, \dots, a_n$  ( $a_i \geq 0$ ).
- Mỗi dòng trong  $q$  dòng tiếp theo chứa số nguyên dương  $i$  ( $1 \leq i \leq n$ ) ứng với truy vấn thứ hạng chiều cao của học sinh đứng ở vị trí  $i$  trong hàng. Số lượng truy vấn không vượt quá  $10^6$ .

**Kết quả:** Ghi ra tập tin văn bản **HRANK.OUT** gồm nhiều dòng, mỗi dòng ứng với câu trả lời cho từng truy vấn trong tập tin dữ liệu vào.

**Ví dụ:**

HRANK . INP	HRANK . OUT
7 4	4
1 0 3 0 2 3	6
7	1
2	5
5	
1	

## XÂY THÁP

Cho  $n$  khối lập phương, khối thứ  $i$  có kích thước  $a_i$ . Người ta sử dụng các khối lập phương để xây tháp bằng cách xếp chồng lên nhau theo nguyên tắc: khối lập phương nằm trên phải có kích thước nhỏ hơn khối nằm bên dưới.

**Yêu cầu:** Sử dụng các khối lập phương theo thứ tự đã cho để xây dựng tháp. Mỗi khối lập phương có thể đặt lên đầu của một trong các tháp hiện có hoặc tạo thành tháp mới. Hãy tìm số tháp ít nhất xây dựng được.

**Dữ liệu:** Vào từ tập tin văn bản **TOWERS.INP**

- Dòng đầu tiên chứa số nguyên  $n (1 \leq n \leq 2 \times 10^5)$  – số lượng khối lập phương.
- Dòng tiếp theo chứa dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$  – kích thước các khối lập phương.

**Kết quả:** Ghi ra tập tin văn bản **TOWERS.OUT** một số nguyên là số tháp ít nhất xây dựng được.

**Ví dụ:**

TOWERS . INP	TOWERS . OUT
5 3 8 2 1 5	2



# LÁ CỜ

Steve đang xây dựng hệ thống nhận dạng cờ. Sau một loạt các phép biến đổi, mẫu nhận dạng được chuẩn hóa thành ma trận ký tự 6×9. Ký tự khác nhau tương ứng với màu khác nhau.

Hệ thống đang ở giai đoạn đầu của quá trình triển khai, vì vậy chỉ mới nhận dạng được cờ theo các mẫu:

<i>CCCCCCCCC</i>	<i>CCCCCCCCC</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>
<i>CCCCCCCCC</i>	<i>CCCCCCCCC</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>
<i>BBBBBBBBB</i>	<i>BBBBBBBBB</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>
<i>BBBBBBBBB</i>	<i>BBBBBBBBB</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>
<i>PPPPPPPPP</i>	<i>CCCCCCCCC</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>
<i>PPPPPPPPP</i>	<i>CCCCCCCCC</i>	<i>ZZZBBBCCC</i>	<i>ZZZAAAZZZ</i>

Nói cách khác, cờ có 3 vạch màu cùng độ rộng, nằm ngang hoặc dọc và vạch ở giữa phải khác màu với vạch ở bên.

Với ma trận 6×9 bất kỳ hệ thống có tính khoảng cách của nó tới cờ. Đó là số lượng ít nhất các ký tự cần thay đổi để nhận được lá cờ.

**Yêu cầu:** Cho ma trận 6×9 chứa các ký tự la-tin in hoa. Hãy xác định khoảng cách của nó tới cờ.

**Dữ liệu:** Vào từ tập tin văn bản **FLAG.INP** gồm 6 dòng, mỗi dòng chứa xâu 9 ký tự la-tin in hoa.

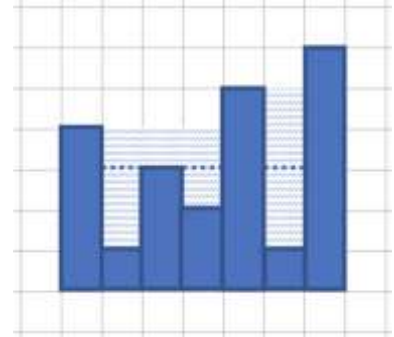
**Kết quả:** Ghi ra tập tin văn bản **FLAG.OUT** một số nguyên – khoảng cách của ma trận tới cờ.

**Ví dụ:**

FLAG . INP	FLAG . OUT
AZZAAAMMA AZZAAAMMA ZZZAMAMMM ZZZAAAMMM AZZAAAMMA AZZAAAMMA	9

**DỰ TRỮ NƯỚC**

Ở quê Bờm có một con mương cung cấp nước cho đồng ruộng. Để điều tiết dòng chảy, người dân đặt  $n$  cột bê tông xếp cạnh nhau có cùng độ rộng với con mương và có chiều cao lần lượt là  $a_1, a_2, \dots, a_n$ .



Khi trời mưa, nước sẽ đọng lại ở các cột có độ cao thấp hơn. Giả sử có 7 cột với độ cao lần lượt là 4, 1, 3, 2, 5, 1, 6 thì lượng nước mưa đọng lại là 10 đơn vị. Để mặt đáy được bằng phẳng, Bờm muốn nâng các cột bê tông lên. Cụ thể chọn một độ cao  $X$  lớn nhất và nâng các cột có độ cao thấp hơn  $X$  lên thành  $X$  mà vẫn đảm bảo lượng nước mưa được giữ lại ít nhất là  $M$  đơn vị.

**Yêu cầu:** Cho độ cao của  $n$  cột bê tông và  $M$  đơn vị nước cần giữ lại. Hãy tìm độ cao  $X$  lớn nhất thỏa mãn yêu cầu. Giả sử lượng nước khi trời mưa là đủ để ngập các vũng nước và không bị thất thoát.

**Dữ liệu:** Vào từ tập tin văn bản **BUILDCOL.INP**

- Dòng đầu tiên gồm hai số nguyên dương  $n$  và  $M$  ( $n \leq 10^5; M \leq 10^{18}$ ).
- Dòng thứ hai gồm  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $a_i \leq 2 \times 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **BUILDCOL.OUT** số nguyên  $X$  thỏa yêu cầu. Nếu không có lời giải thì ghi -1.

**Ví dụ:**

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

## DÃY BAO ĐÓNG

Dãy bao đóng là một dãy con gồm các phần tử liên tiếp sao cho phần tử đầu và phần tử cuối bằng nhau.

Chẳng hạn xét dãy số nguyên 2 2 3 1 2 2 3 1 1, ta có các dãy bao đóng [2 2], [2 2 3 1 2], [1 2 2 3 1], ... Trong đó dãy bao đóng dài nhất có độ dài 6 gồm [2 2 3 1 2 2] và [1 2 2 3 1 1].

**Yêu cầu:** Cho dãy số nguyên  $a_1, a_2, \dots, a_n$ . Hãy tìm dãy bao đóng dài nhất của dãy số.

**Dữ liệu:** Vào từ tập tin văn bản **CLOSURE.INP**

- Dòng đầu tiên chứa số nguyên  $n (1 \leq n \leq 10^6)$ .
- Dòng tiếp theo chứa dãy số nguyên  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^5)$ .

**Kết quả:** Ghi ra tập tin văn bản **CLOUSURE.OUT** một số nguyên là độ dài của bao đóng dài nhất.

**Ví dụ:**

CLOSURE . INP	CLOSURE . OUT
9 2 2 3 1 2 2 3 1 1	6

**NHÀ CAO TÀNG**

Bản đồ nền một khu dự án nhà ở là một hình chữ nhật kích thước  $m \times n$  được chia thành lưới ô vuông đơn vị. Các hàng của lưới được đánh số từ 1 tới  $m$  từ trên xuống dưới và các cột của lưới được đánh số từ 1 tới  $n$  từ trái qua phải. Ô nằm trên giao của hàng  $i$  và cột  $j$  được gọi là ô  $(i, j)$ . Trong bản thiết kế, trên mỗi ô  $(i, j)$  của lưới, người ta muốn xây một tòa nhà hình trụ có chiều cao  $h_{ij}$  và đầy chiếm toàn bộ ô đó.

Từ nóc một tòa nhà, nhìn theo 4 hướng song song với cạnh hình chữ nhật nền, nếu hướng nào cũng bị một tòa nhà khác cao hơn chắn tầm mắt thì tòa nhà đó bị coi là không hợp phong thủy và rất khó bán các căn hộ. Ban quản lý dự án muốn nhờ bạn xác định số lượng những tòa nhà không hợp phong thủy trong thiết kế của dự án.

**Dữ liệu:** Vào từ tập tin văn bản **BUILDING.INP**

- Dòng đầu tiên chứa hai số nguyên dương  $m, n (m, n \leq 1000)$ ,
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa  $n$  số nguyên dương, số thứ  $j$  là  $h_{ij} (h_{ij} \leq 10^6)$

**Kết quả:** Ghi ra tập tin văn bản **BUILDING.OUT** một số nguyên duy nhất là số lượng những tòa nhà không hợp phong thủy trong thiết kế của dự án.

**Ví dụ:**

BUILDING . INP	BUILDING . OUT
4 5 3 2 3 2 3 3 2 <u>1</u> 2 3 1 3 <u>1</u> 3 1 2 2 <u>2</u> 2 2	2