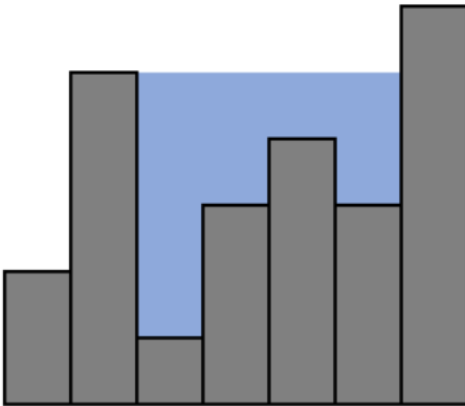


WATER

Tại vương quốc XYZ, có một vật AB vô cùng đặc biệt. AB có N cột đá hình hộp chữ nhật liên tiếp và sát nhau có độ cao lần lượt là h_1, h_2, \dots, h_N . Ta thấy rằng AB có thể chứa được nước. Giả sử $N = 7$, h_i lần lượt bằng 2, 5, 1, 3, 4, 3, 6, ta chứa được lượng nước bằng 9 (như hình vẽ).



Quốc vương muốn chứa nhiều nước nhất có thể. Tuy nhiên, do kinh phí có hạn, quốc vương quyết định tăng thêm 1 đơn vị chiều cao cho nhiều nhất một cột đá bất kì. Hãy giúp quốc vương tính lượng nước tối đa có thể chứa được.

Dữ liệu

Dữ liệu gồm 2 dòng:

- Dòng đầu tiên chứa một số nguyên dương duy nhất N ($N \leq 10^6$).
- Dòng thứ hai chứa N số nguyên dương, số thứ i là độ cao h_i của cột đá thứ i ban đầu ($h_i \leq 10^9$).

Kết quả

- In ra số nguyên duy nhất là kết quả của bài toán.

Ví dụ

Sample Input	Sample Output
7 2 5 1 3 4 3 6	13

Chấm điểm

- Subtask 1 (40% số test): $1 \leq N \leq 10^3$.
- Subtask 2 (60% số test): Không có ràng buộc gì thêm.

CHUỖI NGỌC

Một chuỗi ngọc trai S gồm N hạt được xâu lại với nhau, các hạt được đánh số từ 1 đến N , mỗi hạt có một màu xác định được mã hoá bằng một chữ cái in hoa trong bảng chữ cái tiếng Anh. Hai đoạn ngọc trai trong S là giống nhau khi độ dài của chúng bằng nhau và màu của các cặp hạt tương ứng theo thứ tự xuất hiện trong hai đoạn cũng hoàn toàn giống nhau. Ví dụ hai đoạn AB và AB là giống nhau nhưng AB và BA là không giống nhau.

Hạng của chuỗi ngọc trai S là một số nguyên dương K nhỏ nhất sao cho không tồn tại hai đoạn ngọc trai bất kì giống nhau có cùng độ dài K .

Yêu cầu: Hãy xác định hạng của một chuỗi ngọc trai S .

Dữ liệu: file **chuoingoc.inp**

- Dòng thứ nhất là số nguyên dương N ($1 \leq N \leq 10.000$)
- Dòng thứ hai là chuỗi ngọc trai S độ dài N .

Kết quả: file **chuoingoc.out**

- in ra một số duy nhất là hạng của chuỗi ngọc trai S .

Ràng buộc:

- 50% test: $1 \leq N \leq 100$
- 30% test: $1 \leq N \leq 1000$
- 20% test: theo ràng buộc gốc

Ví dụ:

Chuoingoc.inp	Chuoingoc.out
5 AABAA	3

COMFORTABLE

Cho một bảng ô vuông kích thước vô hạn, các hàng và các cột được đánh số bắt đầu từ 0 tạo thành các ô vuông đơn vị. Ban đầu, tất cả các ô đều rỗng. Khang có N con vật yêu thích và lần lượt đặt chúng vào các ô vuông, con vật thứ i sẽ chiếm 1 ô (x_i, y_i) và không có hai con vật nào vào cùng một ô.

Một con vật được cho là “thỏa mái” nếu nó nằm kề với chính xác 3 con vật khác. Một con vật được gọi là nằm kề với con vật khác nếu ô của chúng có chung cạnh. Thật không may, những con vật quá thỏa mái có xu hướng lười vận động, vì vậy Khang muốn bổ sung thêm những con vật khác cho đến khi không còn con vật nào cảm thấy thỏa mái (kể cả những con mà Khang thêm vào).

Yêu cầu: Với mỗi con vật thứ i ($i=1..N$) sau khi thêm vào, Hãy cho biết số con vật tối thiểu mà Khang phải thêm vào cho đến khi không có con vật nào cảm thấy thỏa mái tình đến thời điểm sau khi thêm con vật thứ i .

Input: file **COMFORTABLE2.INP** gồm:

- Dòng 1: Chứa số nguyên N ($1 \leq N \leq 10^5$);
- N dòng tiếp theo, dòng thứ i chứa hai số nguyên x_i, y_i là tọa độ ô mà Khang đặt con vật thứ i ($i=1..N$), ($0 \leq x_i, y_i \leq 1000$).

Output: file **COMFORTABLE2.OUT** gồm:

- N dòng, dòng thứ i ghi số lượng ít nhất con vật cần thêm sau khi đặt con vật thứ i để cho tất cả các con vật trên bảng không cảm thấy thoải mái. Lưu ý rằng các con vật được thêm vào không nhất thiết phải có tọa độ x và y trong phạm vi $0 \dots 1000$.

Ví dụ:

COMFORTABLE2.INP	COMFORTABLE2.OUT
9	0
0 1	0
1 0	0
1 1	1
1 2	0
2 1	0
2 2	1
3 1	2
3 2	4
4 1	