

Demo Contest 20192

A. SCALAR

1 second, 256 megabytes

Cho hai vector n chiều với tọa độ nguyên: $X = (x_1, x_2, \dots, x_n)$ và $Y = (y_1, y_2, \dots, y_n)$. Tích vô hướng của X và Y là: $XY = \sum_{i=1}^n x_i y_i$. Bạn được quyền đổi chỗ các tọa độ của Y một cách tùy ý và giữ nguyên tọa độ của X . Hãy tìm tích vô hướng nhỏ nhất có thể có. Cụ thể hơn, cần tính giá trị nhỏ nhất của $\sum_{i=1}^n x_i y_{p_i}$ với p là một hoán vị của $\{1, 2, \dots, n\}$

Input

- Dòng đầu chứa số lượng test: T
- $3T$ dòng tiếp theo, cứ ba dòng mô tả một test: Dòng đầu chứa số n , hai dòng sau ghi tọa độ của X và Y

Output

Với test thứ i , in kết quả trên một dòng theo định dạng: Case $\#i$: answer

Scoring

- $1 \leq T, n \leq 1000, |x_i|, |y_i| \leq 10^5$
- Có 50% test với $n \leq 10$

input
2
3
1 3 -5
-2 4 1
5
1 2 3 4 5
1 0 1 0 1
output
Case #1: -25
Case #2: 6

B. BRCOUNT

1 second, 256 megabytes

Từ thời xa xưa, dãy ngoặc được xem như là biểu tượng của tình yêu đôi lứa. Một cách tự nhiên như cách con người tìm ra lửa, người ta trở nên yêu quý các dãy ngoặc đúng

- Xâu rỗng là một dãy ngoặc đúng
- Nếu A là dãy ngoặc đúng thì (A) cũng là dãy ngoặc đúng
- Nếu A,B là 2 dãy ngoặc đúng thì AB cũng là dãy ngoặc đúng

Ví dụ $((()))$ là một dãy ngoặc đúng còn $()()$ thì không. Ngày còn mận nồng ân ái, Hà Lan đã dành tặng cho Ngạn một dãy ngoặc đúng. Qua năm tháng, dãy ngoặc bị mờ đi một số chỗ, ta có thể biểu diễn các vị trí bị mờ bằng ký tự '?', lúc này dãy ngoặc sẽ trông như "((?(?))". Ngạn bối rối vì dãy ngoặc năm nào không còn nguyên vẹn, anh đã xem nó như minh chứng cho tình yêu của hai người. Dù biết có thể có rất nhiều cách khác nhau, Ngạn vẫn muốn khôi phục lại dãy ngoặc của mình. Suy cho cùng, sửa chữa lại dãy ngoặc cũng chính là đang sửa chữa lại các vết thương trong lòng...
Hãy giúp Ngạn đếm số cách khác nhau để thay thế các ký tự '?' thành '(' hoặc ')' sao cho dãy ngoặc thu được là đúng.

Input

Gồm một xâu S chỉ chứa các ký tự '?', '(', ')'

Output

Ghi số cách khác nhau khôi phục dãy ngoặc đúng, chỉ cần in ra số dư của số cách khi chia cho 1000000007

Scoring

- Có 50% số test với $0 < |S| \leq 20$
- Có 30% số test với $20 < |S| \leq 30$

- Có 20% số test với $30 < |S| \leq 1000$

input
((?(?))
output
5

Các cách khôi phục là: (((()))), ((()())), ((()())), ()((())), ()(())

Statement is not available on English language

C. BOUND SUBSEQ

1 second, 256 megabytes

Cho dãy số nguyên a_1, \dots, a_n và hai giá trị nguyên m và M . Hãy đếm số lượng Q các dãy con bao gồm các phần tử liên tiếp của dãy đã cho có tổng các phần tử lớn hơn hoặc bằng m và nhỏ hơn hoặc bằng M .

Input

- Dòng 1: n, m, M ($1 \leq n \leq 20000, 1 \leq m \leq M \leq 100000$)
- Dòng 2: a_1, \dots, a_n ($-20000 \leq a_i \leq 20000$)

Output

Ghi ra giá trị Q

input
5 3 6
5 4 3 2 1
output
6

Có một nửa số test có giá trị n từ 1 đến 1000 và một nửa số test có giá trị n lớn hơn 1000.

D. DOI

1 second, 256 megabytes

Cho dãy số nguyên a_1, a_2, \dots, a_n . Ta gọi dãy con gồm các phần tử liên tiếp của dãy đã cho $a_i, \dots, a_j, \dots, a_k$ ($1 \leq i < j < k \leq n$) là đồi nếu $a_t < a_{t+1}$ với mọi $i \leq t < j$ và $a_t > a_{t+1}$ với mọi $j \leq t < k$. Trong trường hợp này ta gọi $\min\{j - i, k - j\}$ là độ cao của đồi. Tương tự như vậy, ta gọi dãy con gồm các phần tử liên tiếp của dãy đã cho $a_i, \dots, a_j, \dots, a_k$ ($1 \leq i < j < k \leq n$) là thung lũng nếu $a_t > a_{t+1}$ với mọi $i \leq t < j$ và $a_t < a_{t+1}$ với mọi $j \leq t < k$. Trong trường hợp này ta gọi $\min\{j - i, k - j\}$ là độ sâu của thung lũng.

Yêu cầu: Hãy tính độ cao của đồi cao nhất và độ sâu của thung lũng sâu nhất của dãy số đã cho.

Input

- Dòng thứ nhất chứa số nguyên n ($1 \leq n \leq 10^6$) là số lượng phần tử của dãy số
- Dòng thứ hai chứa n số nguyên là các phần tử của dãy số (hai số liên tiếp được ghi cách nhau bởi dấu cách. Mỗi phần tử của dãy số có trị tuyệt đối không vượt quá 100000).

Output

Một dòng chứa hai số nguyên là độ cao của đồi cao nhất và độ sâu của thung lũng sâu nhất. Nếu không có đồi hoặc không có thung lũng thì vị trí tương ứng ghi số 0.

input
10 4 4 1 6 3 2 1 2 5 7
output
1 3

input
10 2 3 4 5 6 7 8 9 10 9
output
1 0