

ESTIMATION

Một trong những kĩ thuật thường được sử dụng trong các cuộc thi lập trình Marathon để tìm lời giải gần tối ưu cho các bài toán NP là kĩ thuật Leo đồi. Tuy nhiên, kĩ thuật leo đồi không phải kĩ thuật tối ưu để tìm lời giải cho mọi bài toán NP. Trong một số bài toán, hàm mục tiêu (hàm đánh giá độ tốt của lời giải) có thể chứa nhiều đồng bằng (đồng bằng là các khoảng mà giá trị của hàm mục tiêu ở đó là hằng số). Nếu rơi vào trường hợp đồng bằng, kĩ thuật Leo đồi sẽ không có hướng để tối ưu và gặp khó khăn trong việc tìm một lời giải tốt. Khi đó, các bạn sẽ phải sử dụng tiếp kĩ thuật "khử đồng bằng" hoặc chọn kĩ thuật khác thay cho Leo đồi (ví dụ như Giải thuật Luyện kim).

Để kiểm tra khả năng "khử đồng bằng" của mình, Kc97 quyết định sử dụng kĩ thuật Leo đồi cho bài toán sau:

Cho dãy số A gồm n số nguyên, tìm giá trị của số lớn nhất và số nhỏ nhất trong dãy.

Để dãy A giống một hàm số ngoài đời thực, Kc97 đã sinh nhiều dãy A , mỗi dãy có một quy luật khác nhau như dãy $A_x = x^2 + 2x - 1$, $A_x = |x|$, Nhiệm vụ của bạn bây giờ là giúp Kc97 tạo ra k đồng bằng trong mảng A , tức là biến mảng A có n số nguyên thành mảng B có n số nguyên sao cho mảng B có k đoạn phần tử bằng nhau liên tiếp, mỗi phần tử trong mảng thuộc một đoạn và $\sum_{i=1}^n |A_i - B_i|$ đạt giá trị nhỏ nhất (điều kiện này nhằm đảm bảo mảng B vẫn giống một hàm số ngoài đời thực).

Dữ liệu

- Dòng đầu tiên gồm hai số nguyên n và k ($1 \leq n \leq 2000, 1 \leq k \leq n, k \leq n$) lần lượt là độ dài mảng A và số đồng bằng bạn cần tạo.
- n dòng tiếp theo, dòng thứ i chứa một số nguyên thuộc đoạn $[-10000; 10000]$ là số nguyên thứ i trong mảng A .

Kết quả

- Gồm một dòng chứa một số nguyên duy nhất là tổng $\sum_{i=1}^n |A_i - B_i|$ của mảng B bạn tạo được từ mảng A .

Ví dụ

Sample Input	Sample Output
7 2 6 5 4 3 2 1 7	9