



# Energy exchange

Link submit: <http://codeforces.com/problemset/problem/68/B>

Solution:

C++	<a href="https://ideone.com/QhWrnE">https://ideone.com/QhWrnE</a>
Java	<a href="https://ideone.com/SHBnZP">https://ideone.com/SHBnZP</a>
Python	<a href="https://ideone.com/8MiDOC">https://ideone.com/8MiDOC</a>

Tóm tắt đề:

Little muốn các accumulator chứa lượng năng lượng là như nhau. Ban đầu, mỗi accumulator có một lượng năng lượng:  $i$ -th accumulator có  $a_i$  năng lượng. Năng lượng có thể chuyển từ accumulator này sang accumulator khác. Mỗi lần  $x$  năng lượng được chuyển đi thì sẽ bị giảm đi  $k\%$ . Tức là, năng lượng được chuyển từ một accumulator sang accumulator khác, năng lượng của accumulator ban đầu giảm đi  $x$ , và accumulator khác sẽ tăng  $x - \frac{xk}{100}$  đơn vị.

Tìm giá trị lớn nhất sao cho lượng năng lượng chứa ở mỗi thùng là bằng nhau sau khi chuyển.

Input:

Dòng đầu tiên chứa hai số nguyên  $n$  và  $k$  ( $1 \leq n \leq 10000, 0 \leq k \leq 99$ ) là số lượng accumulator và phần trăm lượng năng lượng sẽ bị mất khi chuyển. Dòng tiếp theo chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  là cho biết năng lượng của mỗi accumulator ( $0 \leq a_i \leq 1000, 1 \leq i \leq n$ ).

Output:

In ra lượng năng lượng lớn nhất có thể được chứa trong mỗi accumulator sau khi chuyển.

Ví dụ:

3 50	2.0000000000
4 2 1	

Giải thích ví dụ:

Nhìn vào có thể thấy accumulator 1 chuyển 2 năng lượng qua accumulator 3:

- Accumulator 1 còn 2 đơn vị.
- Lượng chuyển đi giảm 50%, nên accumulator 3 nhận 1  $\Rightarrow$  accumulator 3 chứa 2 đơn vị.

$\Rightarrow$  Tất cả các accumulator đều có 2 đơn vị.

### Hướng dẫn giải:

Gọi  $SumEnergy$  là tổng năng lượng của các accumulator ban đầu.

Gọi  $m$  là lượng năng lượng ở mỗi accumulator, vậy với  $n$  accumulator thì sẽ có tổng năng lượng là  $m * n$ .

Những accumulator có lượng năng lượng lớn hơn  $m$  sẽ được chuyển đi, tức sẽ chuyển đi  $x_i = a_i - m$  đơn vị (năng lượng không lớn hơn  $m$  thì  $x_i = 0$ ), vậy gọi tổng năng lượng được chuyển đi là  $SumTransfer = (x_i + x_{i+1} + \dots + x_n)$ .

Với mỗi  $x_i$  ở mỗi thùng chuyển đi sẽ mất  $\frac{x_i k}{100}$  năng lượng, vậy tổng  $x_i, x_{i+1}, \dots, x_n$  chuyển đi sẽ mất  $\frac{x_i k}{100} + \frac{x_{i+1} k}{100} + \dots + \frac{x_n k}{100} = \frac{(x_i + x_{i+1} + \dots + x_n) k}{100} = \frac{SumTransfer * k}{100}$ , gọi tổng lượng mất là  $SumLost$ .

Tổng năng lượng sau khi chuyển sẽ bằng tổng năng lượng ban đầu trừ đi số lượng bị mất.

Vậy trường hợp may mắn nhất là  $m * n = SumEnergy - SumLost$ .

Bây giờ, đi tìm  $m$  sao cho độ chênh lệch giữa 2 vế là nhỏ nhất có thể. Ta có thể thấy,  $m$  chắc chắn không thể nào lớn hơn  $\max(a_i)$  và cũng không thể nào nhỏ hơn  $\min(a_i)$ .

- Bước 1: Tính tổng  $SumEnergy$ .
- Bước 2: Dùng binary search để tìm  $m$ .
  - Đặt giá trị ban đầu  $left = \min(a_i), right = \max(a_i)$ , mà  $0 \leq a_i \leq 1000$ , vậy nên ta cũng có thể đặt giá trị ban đầu  $left = 0, right = 1000$ .
  - Trong khi  $left \leq right$ :
    - Tính  $mid = (left + right) / 2$
    - Tính  $SumEnergy, SumLost$  dựa vào  $mid$  vừa tính được ở trên
    - Trường hợp  $mid * n < SumEnergy - SumLost$  thì  $left = mid$
    - Ngược lại,  $right = mid$
  - Giá trị  $mid$  có thể không là số nguyên, tức ta đang chặt nhị phân trên số thực. Mà thực tế thì việc chặt nhị phân trên số thực đến khi  $left > right$  là điều không thể, vậy nên chỗ kiểm tra ta thay bằng  $right - left > 1e - 7$  (chọn độ chênh lệch giữa  $left, right$  sao cho đủ nhỏ để kết quả trả về so với đáp án của đề không chênh lệch quá  $10^{-6}$ ).
- Giá trị cần tìm là  $mid$  cuối cùng nhận được.

**Độ phức tạp:**  $O(N * \log(1e7))$  với  $N$  là số lượng accumulator.