## Energy Exchange

### Input

N, K: N <= 10^4, 0 <= K <= 99

a[i]: a[i] <= 1000

### Output

maximum energy in each accumulators

relative error not exceed 10^-6

### Sample test:

n = 3, k = 50%

a b c

4 2 1

di chuyển 2 năng lượng thùng a sang thùng c

2 năng lượng -> bị mất 1 năng lượng -> thùng c chỉ nhận thêm 1 năng lượng

thùng a: 4 - 2 = 2

thùng b: 2

thùng c: 1 + 1 = 2

=> output: 2

x - x\*k/100 = x \* (100 - k)/100

ví dụ k = 90, lượng nhaanj: x \* (100 - 90)/100 = x \* 10/100 = x \* 10%

n = 2, k = 90

a b

1 11

di chuyển (11-1.9099090) từ bình b sang bình a

bình a: 1,90909

bình b: 1 + (11-1.9099090) \* 10% = 1,909090

### Ideas

Chặt nhị phân với số thực:

cách 1: 2 số x và y bằng nhau nếu như chênh lệch không quá 10^-8

1,00000001 == 1,000000009

0, 000000001 => 0

.....

1000,000000001 => 10^13

log(10^13)

độ phức tạp O(log(10^13) \* N)

cách 2: thực hiện so sánh x lần (x = 100, 1000)

// while(left <= right)

for(int dem = 1; dem <= 100, 1000, ..; ++dem) {

mid = ....

if (....) left = mid

else .....

}

### Code

// kiểm tra có thể phân phối năng lượng sao cho mỗi bình chứa ít nhất x năng lượng

function check(double x) {

double needTransfer = 0;

double canTransfer = 0;

for(i: 1->n) {

if(a[i] >= x) canTransfer += (a[i] - x);

else needTransfer += (x - a[i]);

}

return canTransfer \* (100-k)/100 >= needTransfer;

}

function main() {

input(n, k);

for(i: 1 -> n) {

input(a[i]);

}

float left = 0, right = 1000, mid;

for(dem: 1->1000) {

mid = (left + right) / 2;

if(check(mid)) {

left = mid;

} else {

right = mid;

}

}

output(left);

}

độ phức tạp O(số lần chặt nhị phân \* đpt kiểm tra) = O(1000 \* N)