

## 10647 Optimal House Placement

Mr. G. Enerous has a lot of friends. All his friends are living in the same street. Now every year at christmas Mr. Enerous has the problem to bring parcels to all his friends. To make parcel delivering easier for him he decides to move house to this street.

When Mr. G. Enerous is bearing a parcel from one house to another and the distance between these houses is  $x$ , he consumes  $x * x$  energy units. When he doesn't bear a parcel, the energy consumption is negligible. Since the parcels are very big, he can bear only one parcel at a time, and he doesn't want to stop outside houses because he doesn't want that the parcels become filthy. So if he is delivering a parcel to one of his friends, he starts at his house and goes in the direction of the house of his friend, but to minimize energy consumption he stops at each house of a friend on the way.

Can you tell him where in the street he should place his house so that the energy required to deliver all the parcels is minimized?

You may assume that for any reasonable house position for Mr. Enerous' house the energy required to deliver all the parcels rounded to the next integer fits into a signed 64 bit integer. Here a reasonable house position is any position between smallest house coordinate and biggest house coordinate.

### Input

The input consists of several test cases. The first line of each test case contains a positive integer  $n$  ( $2 \leq n \leq 10000$ ), the number of friends of Mr. Enerous. In the next  $n$  lines you are given the position of the houses in increasing order. Each position of a house is a non negative integer  $< 1000000$ .

The end of input is signaled by a test case with  $n = 0$ .

### Output

For each test case write one floating point number with exactly 3 digits after the decimal point that is the position where Mr. Enerous should place his house in order to minimize the energy required to deliver all the parcels. If the exact value of the minimum energy can be reached with several house positions, print the smallest one.

### Sample Input

```
3
1
2
3
4
1
2
4
5
4
0
1
2
999999
0
```

**Sample Output**

1.667

3.000

250001.250