A. Jeff and Rounding

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Jeff got $2n$ real numbers $a_1, a_2, ..., a_{2n}$ as a birthday present. The boy hates non-integer numbers, so he decided to slightly "adjust" the numbers he's got. Namely, Jeff consecutively executes n operations, each of them goes as follows:

- choose indexes i and j $(i \neq j)$ that haven't been chosen yet;
- round element a_i to the nearest integer that isn't more than a_i (assign to a_i: $\lfloor a_i \rfloor$);
- round element a_j to the nearest integer that isn't less than a_j (assign to a_j: $\lceil a_j \rceil$).

Nevertheless, Jeff doesn't want to hurt the feelings of the person who gave him the sequence. That's why the boy wants to perform the operations so as to make the absolute value of the difference between the sum of elements before performing the operations and the sum of elements after performing the operations as small as possible. Help Jeff find the minimum absolute value of the difference.

Input

The first line contains integer n $(1 \le n \le 2000)$. The next line contains $2n$ real numbers $a_1, a_2, ..., a_{2n}$ $(0 \le a_i \le 10000)$, given with exactly three digits after the decimal point. The numbers are separated by spaces.

Output

In a single line print a single real number — the required difference with **exactly three digits** after the decimal point.

Examples

input

```
3
0.000 0.500 0.750 1.000 2.000 3.000
```

output

```
0.250
```

input

```
3
4469.000 6526.000 4864.000 9356.383 7490.000 995.896
```

output

```
0.279
```

Note

In the first test case you need to perform the operations as follows: $(i = 1, j = 4)$, $(i = 2, j = 3)$, $(i = 5, j = 6)$. In this case, the difference will equal $|(0 + 0.5 + 0.75 + 1 + 2 + 3)$ - $(0 + 0 + 1 + 1 + 2 + 3)| = 0.25$.