

PDP-17 (2005) - Phase C' - 2 (land)

Time limit: 15.0s Memory limit: 64M

Property Union

In a plot of land, there are N owners, where each owner possesses a portion of the land by a percentage X_i , for $1 \leq i \leq N$. The value of X_i is a real number representing the corresponding ownership percentage. Obviously, the percentages of all owners sum up to 1 (i.e., $\sum X_i = 1$). The owners want to merge their properties into a single unified property. However, this process can only be done step by step. In other words, in the first phase, two properties will be merged to form a new property. Thus, a total of $N - 1$ properties will be created. Then, from these $N - 1$ properties, two will be selected to merge together until we reach a final state where two properties will merge to form a single plot of land. It is assumed that two properties can be merged regardless of whether they are adjacent or not.

It is given that the cost of merging two properties i and j is equal to $X_i + X_j$, the total cost depends on the order of merging the properties. The problem lies in finding the minimum total cost for merging the plots into a single unified property. To achieve this, you need to construct a program named `land.c` or `land.pas` depending on the programming language used, with the following characteristics.

Input

The file `land.in` contains an integer on the first line representing the number of owners N , where $1 \leq N \leq 50\,000$. Each of the next N lines contains a real number X representing the ownership percentage of each owner.

Output

The file `land.out` consists of a real number rounded to two decimal places, indicating the minimum merging cost.

Example of Input - Output Files:

1st

STDIN (`land.in`)

```
3
0.10
0.385
0.515
```

STDOUT (`land.out`)

```
1.49
```

2nd

STDIN (land.in)

```
4
0.17
0.22
0.23
0.38
```

STDOUT (land.out)

```
2.00
```

Time Limit: 15 sec per test.

Clarification: Each line, both in the input and output files, ends with a newline character (for C and C++ programmers: "\n", while for Pascal programmers the use of ReadLn and WriteLn is sufficient).