A. Jeff and Digits

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

Jeff's got n cards, each card contains either digit 0, or digit 5. Jeff can choose several cards and put them in a line so that he gets some number. What is the largest possible number divisible by 90 Jeff can make from the cards he's got?

Jeff must make the number without leading zero. At that, we assume that number 0 doesn't contain any leading zeroes. Jeff doesn't have to use all the cards.

Input

The first line contains integer n $(1 \le n \le 10^3)$. The next line contains n integers a_1, a_2, ..., a_n ($a_i = 0$ or $a_i = 5$). Number a_i represents the digit that is written on the i-th card.

Output

In a single line print the answer to the problem — the maximum number, divisible by 90. If you can't make any divisible by 90 number from the cards, print -1.

Sample test(s)

input

```
4
5 0 5 0
```

output

```
0
```

input

```
11
5 5 5 5 5 5 5 5 0 5 5
```

output

```
5555555550
```

Note

In the first test you can make only one number that is a multiple of 90 — 0.

In the second test you can make number 5555555550, it is a multiple of 90.

B. Jeff and Periods

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

One day Jeff got hold of an integer sequence a_1, a_2, ..., a_n of length n. The boy immediately decided to analyze the sequence. For that, he needs to find all values of x, for which these conditions hold:

- x occurs in sequence a.
- Consider all positions of numbers x in the sequence a (such i, that $a_i = x$). These numbers, sorted in the increasing order, must form an arithmetic progression.

Help Jeff, find all x that meet the problem conditions.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The next line contains integers a_1, a_2, ..., a_n $(1 \le a_i \le 10^5)$. The numbers are separated by spaces.

Output

In the first line print integer t — the number of valid x. On each of the next t lines print two integers x and p_x, where x is current suitable value, p_x is the common difference between numbers in the progression (if x occurs exactly once in the sequence, p_x must equal 0). Print the pairs in the order of increasing x.

Sample test(s)

```
input
```

```
1
2
```

```
output
```

```
1
2 0
```

```
input
```

```
8
1 2 1 3 1 2 1 5
```

```
output
```

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

Note

In the first test 2 occurs exactly once in the sequence, ergo $p_2 = 0$.

C. 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.

Sample test(s)

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 $|(0 + 0.5 + 0.75 + 1 + 2 + 3) - (0 + 0 + 1 + 1 + 2 + 3)| = 0.25$.

D. Jeff and Furik

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

Jeff has become friends with Furik. Now these two are going to play one quite amusing game.

At the beginning of the game Jeff takes a piece of paper and writes down a permutation consisting of n numbers: $p_1, p_2, ..., p_n$. Then the guys take turns to make moves, Jeff moves first. During his move, Jeff chooses two adjacent permutation elements and then the boy swaps them. During his move, Furic tosses a coin and if the coin shows "heads" he chooses a random pair of adjacent elements with indexes i and $i+1$, for which an inequality $p_i > p_{i+1}$ holds, and swaps them. But if the coin shows "tails", Furik chooses a random pair of adjacent elements with indexes i and $i+1$, for which the inequality $p_i < p_{i+1}$ holds, and swaps them. If the coin shows "heads" or "tails" and Furik has multiple ways of adjacent pairs to take, then he uniformly takes one of the pairs. If Furik doesn't have any pair to take, he tosses a coin one more time. The game ends when the permutation is sorted in the increasing order.

Jeff wants the game to finish as quickly as possible (that is, he wants both players to make as few moves as possible). Help Jeff find the minimum mathematical expectation of the number of moves in the game if he moves optimally well.

You can consider that the coin shows the heads (or tails) with the probability of 50 percent.

Input

The first line contains integer n $(1 \le n \le 3000)$. The next line contains n distinct integers $p_1, p_2, ..., p_n$ $(1 \le p_i \le n)$ — the permutation p. The numbers are separated by spaces.

Output

In a single line print a single real value — the answer to the problem. The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Sample test(s)

input

```
2
1 2
```

output

```
0.000000
```

input

```
5
3 5 2 4 1
```

output

```
13.000000
```

Note

In the first test the sequence is already sorted, so the answer is 0.

E. Jeff and Brackets

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

Jeff loves regular bracket sequences.

Today Jeff is going to take a piece of paper and write out the regular bracket sequence, consisting of nm brackets. Let's number all brackets of this sequence from 0 to $nm - 1$ from left to right. Jeff knows that he is going to spend $a_{i \bmod n}$ liters of ink on the i-th bracket of the sequence if he paints it opened and $b_{i \bmod n}$ liters if he paints it closed.

You've got sequences a, b and numbers n, m. What minimum amount of ink will Jeff need to paint a regular bracket sequence of length nm?

Operation $x \bmod y$ means taking the remainder after dividing number x by number y.

Input

The first line contains two integers n and m $(1 \le n \le 20;\ 1 \le m \le 10^7;\ m$ is even). The next line contains n integers: $a_0, a_1, ..., a_{n-1}$ $(1 \le a_i \le 10)$. The next line contains n integers: $b_0, b_1, ..., b_{n-1}$ $(1 \le b_i \le 10)$. The numbers are separated by spaces.

Output

In a single line print the answer to the problem — the minimum required amount of ink in liters.

Sample test(s)

input

```
2 6
1 2
2 1
```

output

```
12
```

input

```
1 10000000
2
3
```

output

```
25000000
```

Note

In the first test the optimal sequence is: $()()()()()()$, the required number of ink liters is 12.