A. Book Reading

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Luba bought a very interesting book. She knows that it will take t seconds to read the book. Luba wants to finish reading as fast as she can.

But she has some work to do in each of n next days. The number of seconds that Luba has to spend working during i-th day is a_i. If some free time remains, she can spend it on reading.

Help Luba to determine the minimum number of day when she finishes reading.

It is guaranteed that the answer doesn't exceed n.

Remember that there are 86400 seconds in a day.

Input

The first line contains two integers n and t $(1 \le n \le 100, 1 \le t \le 10^6)$ — the number of days and the time required to read the book.

The second line contains n integers a_i $(0 \le a_i \le 86400)$ — the time Luba has to spend on her work during i-th day.

Output

Print the minimum day Luba can finish reading the book.

It is guaranteed that answer doesn't exceed n.

Examples

input

```
2 2
86400 86398
```

output

```
2
```

input

```
2 86400
0 86400
```

output

```
1
```

B. Japanese Crosswords Strike Back

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

A one-dimensional Japanese crossword can be represented as a binary string of length x. An encoding of this crossword is an array a of size n, where n is the number of segments formed completely of 1's, and a_i is the length of i-th segment. No two segments touch or intersect.

For example:

- If $x = 6$ and the crossword is 111011, then its encoding is an array $\{3, 2\}$;
- If $x = 8$ and the crossword is 01101010, then its encoding is an array $\{2, 1, 1\}$;
- If $x = 5$ and the crossword is 11111, then its encoding is an array $\{5\}$;
- If $x = 5$ and the crossword is 00000, then its encoding is an empty array.

Mishka wants to create a new one-dimensional Japanese crossword. He has already picked the length and the encoding for this crossword. And now he needs to check if there is **exactly one** crossword such that its length and encoding are equal to the length and encoding he picked. Help him to check it!

Input

The first line contains two integer numbers n and x ($1 \le n \le 100000$, $1 \le x \le 10^9$) — the number of elements in the encoding and the length of the crossword Mishka picked.

The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10000$) — the encoding.

Output

Print `YES` if there exists **exaclty one** crossword with chosen length and encoding. Otherwise, print `NO`.

Examples

input

```
2 4
1 3
```

output

```
NO
```

input

```
3 10
3 3 2
```

output

```
YES
```

input

```
2 10
1 3
```

output

```
NO
```

C. Bertown Subway

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

The construction of subway in Bertown is almost finished! The President of Berland will visit this city soon to look at the new subway himself.

There are n stations in the subway. It was built according to the *Bertown Transport Law*:

1. For each station i there exists exactly one train that goes from this station. Its destination station is p_i, possibly $p_i = i$;
2. For each station i there exists exactly one station j such that $p_j = i$.

The President will consider the *convenience* of subway after visiting it. The *convenience* is the number of ordered pairs (x, y) such that person can start at station x and, after taking some subway trains (possibly zero), arrive at station y $(1 \le x, y \le n)$.

The mayor of Bertown thinks that if the subway is not *convenient* enough, then the President might consider installing a new mayor (and, of course, the current mayor doesn't want it to happen). Before President visits the city mayor has enough time to rebuild some paths of subway, thus changing the values of p_i for **not more than two subway stations**. Of course, breaking the *Bertown Transport Law* is really bad, so the subway must be built according to the *Law* even after changes.

The mayor wants to do these changes in such a way that the *convenience* of the subway is maximized. Help him to calculate the maximum possible *convenience* he can get!

Input

The first line contains one integer number n $(1 \le n \le 100000)$ — the number of stations.

The second line contains n integer numbers $p_1, p_2, ..., p_n$ $(1 \le p_i \le n)$ — the current structure of the subway. All these numbers are distinct.

Output

Print one number — the maximum possible value of *convenience*.

Examples

input

```
3
2 1 3
```

output

```
9
```

input

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

output

```
17
```

Note

In the first example the mayor can change p_2 to 3 and p_3 to 1, so there will be 9 pairs: $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(3, 1)$, $(3, 2)$, $(3, 3)$.

In the second example the mayor can change p_2 to 4 and p_3 to 5.

D. Boxes And Balls

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ivan has n different boxes. The first of them contains some balls of n different colors.

Ivan wants to play a strange game. He wants to distribute the balls into boxes in such a way that for every i $(1 \le i \le n)$ i-th box will contain all balls with color i.

In order to do this, Ivan will make some turns. Each turn he does the following:

1. Ivan chooses any non-empty box and takes all balls from this box;
2. Then Ivan chooses any k empty boxes (the box from the first step becomes empty, and Ivan is allowed to choose it), separates the balls he took on the previous step into k non-empty groups and puts each group into one of the boxes. He should put each group into a separate box. He can choose either $k = 2$ or $k = 3$.

The *penalty* of the turn is the number of balls Ivan takes from the box during the first step of the turn. And *penalty* of the game is the total *penalty* of turns made by Ivan until he distributes all balls to corresponding boxes.

Help Ivan to determine the minimum possible *penalty* of the game!

Input

The first line contains one integer number n $(1 \le n \le 200000)$ — the number of boxes and colors.

The second line contains n integer numbers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$, where a_i is the number of balls with color i.

Output

Print one number — the minimum possible *penalty* of the game.

Examples

input

```
3
1 2 3
```

output

```
6
```

input

```
4
2 3 4 5
```

output

```
19
```

Note

In the first example you take all the balls from the first box, choose $k = 3$ and sort all colors to corresponding boxes. Penalty is 6.

In the second example you make two turns:

1. Take all the balls from the first box, choose $k = 3$, put balls of color 3 to the third box, of color 4 — to the fourth box and the rest put back into the first box. Penalty is 14;
2. Take all the balls from the first box, choose $k = 2$, put balls of color 1 to the first box, of color 2 — to the second box. Penalty is 5.

Total penalty is 19.

E. Binary Matrix

time limit per test: 3 seconds
memory limit per test: 16 megabytes
input: standard input
output: standard output

You are given a matrix of size $n \times m$. Each element of the matrix is either 1 or 0. You have to determine the number of connected components consisting of 1's. Two cells belong to the same component if they have a common border, and both elements in these cells are 1's.

Note that the memory limit is unusual!

Input

The first line contains two numbers n and m ($1 \le n \le 2^{12}$, $4 \le m \le 2^{14}$) — the number of rows and columns, respectively. It is guaranteed that m is divisible by 4.

Then the representation of matrix follows. Each of n next lines contains $\frac{m}{4}$ one-digit hexadecimal numbers (that is, these numbers can be represented either as digits from 0 to 9 or as uppercase Latin letters from A to F). Binary representation of each of these numbers denotes next 4 elements of the matrix in the corresponding row. For example, if the number B is given, then the corresponding elements are `1011`, and if the number is 5, then the corresponding elements are `0101`.

Elements are not separated by whitespaces.

Output

Print the number of connected components consisting of 1's.

Examples

input

```
3 4
1
A
8
```

output

```
3
```

input

```
2 8
5F
E3
```

output

```
2
```

input

```
1 4
0
```

output

```
0
```

Note

In the first example the matrix is:

```
0001
1010
1000
```

It is clear that it has three components.

The second example:

```
01011111
11100011
```

It is clear that the number of components is 2.

There are no 1's in the third example, so the answer is 0.

F. Anti-Palindromize

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A string a of length m is called *antipalindromic* iff m is even, and for each i ($1 \le i \le m$) $a_i \ne a_{m-i+1}$.

Ivan has a string s consisting of n lowercase Latin letters; n is even. He wants to form some string t that will be an *antipalindromic* permutation of s. Also Ivan has denoted the *beauty* of index i as b_i, and the *beauty* of t as the sum of b_i among all indices i such that $s_i = t_i$.

Help Ivan to determine maximum possible *beauty* of t he can get.

Input

The first line contains one integer n ($2 \le n \le 100$, n is even) — the number of characters in s.

The second line contains the string s itself. It consists of only lowercase Latin letters, and it is guaranteed that its letters can be reordered to form an *antipalindromic* string.

The third line contains n integer numbers b_1, b_2, ..., b_n ($1 \le b_i \le 100$), where b_i is the *beauty* of index i.

Output

Print one number — the maximum possible *beauty* of t.

Examples

input

```
8
abacabac
1 1 1 1 1 1 1 1
```

output

```
8
```

input

```
8
abaccaba
1 2 3 4 5 6 7 8
```

output

```
26
```

input

```
8
abacabca
1 2 3 4 4 3 2 1
```

output

```
17
```