A. Interesting Signs

time limit per test: 2 seconds
memory limit per test: 256 megabytes

You just finished your shift volunteering at Swinburne University Open Day. Well, you are not quite finished yet! Your job now is to take down some of the giant signs that were set up around campus. A sign consists of a sequence of upper-case English letters.

Since the letters are huge, you are only strong enough to take down and carry two of the letters at a time. To stop the sign from falling over, you may only take letters from either end of the sign. For efficiency reasons, your boss requires that you only ever carry two of the same letter at a time, since they will be stored next to each other in storage. Finally, you must always be carrying two letters (unless there is only one letter left), otherwise your boss will think that you are lazy. Note that you may take two letters from the same end of the sign, or one from each end. You are not sure that this will always be possible for every sign, so you are going to write a computer program to find out.

Input

The input contains a single line with one string s, which is the sign to be taken down. The length of s is between 1 and $1\,000$, inclusive, and contains only uppercase English characters.

Output

Display if it is possible to take down the sign.

Examples

input	output
SWINBURNE	IMPOSSIBLE

input	output
RACECAR	POSSIBLE

input	output
VCPC	IMPOSSIBLE

input	output
ACCIDIA	POSSIBLE

input	output
AARRGHH	POSSIBLE

B. Hard problem

time limit per test: 1 second
memory limit per test: 256 megabytes

Vasiliy is fond of solving different tasks. Today he found one he wasn't able to solve himself, so he asks you to help.

Vasiliy is given n strings consisting of lowercase English letters. He wants them to be sorted in lexicographical order (as in the dictionary), but he is not allowed to swap any of them. The only operation he is allowed to do is to reverse any of them (first character becomes last, second becomes one before last and so on).

To reverse the i-th string Vasiliy has to spent c_i units of energy. He is interested in the minimum amount of energy he has to spent in order to have strings sorted in lexicographical order.

String A is lexicographically smaller than string B if it is shorter than B ($|A| < |B|$) and is its prefix, or if none of them is a prefix of the other and at the first position where they differ character in A is smaller than the character in B.

For the purpose of this problem, two equal strings nearby do not break the condition of sequence being sorted lexicographically.

Input

The first line of the input contains a single integer n ($2 \le n \le 100\,000$) — the number of strings.

The second line contains n integers c_i ($0 \le c_i \le 10^9$), the i-th of them is equal to the amount of energy Vasiliy has to spent in order to reverse the i-th string.

Then follow n lines, each containing a string consisting of lowercase English letters. The total length of these strings doesn't exceed $100\,000$.

Output

If it is impossible to reverse some of the strings such that they will be located in lexicographical order, print -1. Otherwise, print the minimum total amount of energy Vasiliy has to spent.

Examples

input
```
2
1 2
ba
ac
```
output
```
1
```

input
```
3
1 3 1
aa
ba
ac
```
output
```
1
```

input
```
2
5 5
bbb
aaa
```
output
```
-1
```

input
```
2
3 3
aaa
aa
```
output
```
-1
```

Note

In the second sample one has to reverse string 2 or string 3. To amount of energy required to reverse the string 3 is smaller.

In the third sample, both strings do not change after reverse and they go in the wrong order, so the answer is -1.

In the fourth sample, both strings consists of characters 'a' only, but in the sorted order string "aa" should go before string "aaa", thus the answer is -1.

C. Coloring Trees

time limit per test: 2 seconds
memory limit per test: 256 megabytes

ZS the Coder and Chris the Baboon has arrived at Udayland! They walked in the park where n trees grow. They decided to be naughty and color the trees in the park. The trees are numbered with integers from 1 to n from left to right.

Initially, tree i has color c_i. ZS the Coder and Chris the Baboon recognizes only m different colors, so $0 \le c_i \le m$, where $c_i = 0$ means that tree i is *uncolored*.

ZS the Coder and Chris the Baboon decides to color only the uncolored trees, i.e. the trees with $c_i = 0$. They can color each of them them in any of the m colors from 1 to m. Coloring the i-th tree with color j requires exactly $p_{i,j}$ litres of paint.

The two friends define the *beauty* of a coloring of the trees as the **minimum** number of contiguous groups (each group contains some subsegment of trees) you can split all the n trees into so that each group contains trees of the same color. For example, if the colors of the trees from left to right are $2, 1, 1, 1, 3, 2, 2, 3, 1, 3$, the beauty of the coloring is 7, since we can partition the trees into 7 contiguous groups of the same color : $\{2\}, \{1, 1, 1\}, \{3\}, \{2, 2\}, \{3\}, \{1\}, \{3\}$.

ZS the Coder and Chris the Baboon wants to color all uncolored trees so that the beauty of the coloring is **exactly** k. They need your help to determine the minimum amount of paint (in litres) needed to finish the job.

Please note that the friends can't color the trees that are already colored.

Input

The first line contains three integers, n, m and k ($1 \le k \le n \le 100$, $1 \le m \le 100$) — the number of trees, number of colors and beauty of the resulting coloring respectively.

The second line contains n integers $c_1, c_2, ..., c_n$ ($0 \le c_i \le m$), the initial colors of the trees. c_i equals to 0 if the tree number i is uncolored, otherwise the i-th tree has color c_i.

Then n lines follow. Each of them contains m integers. The j-th number on the i-th of them line denotes $p_{i,j}$ ($1 \le p_{i,j} \le 10^9$) — the amount of litres the friends need to color i-th tree with color j. $p_{i,j}$'s are specified even for the initially colored trees, but such trees still can't be colored.

Output

Print a single integer, the minimum amount of paint needed to color the trees. If there are no valid tree colorings of beauty k, print -1.

Examples

input	output
3 2 2 0 0 0 1 2 3 4 5 6	10

input	output
3 2 2 2 1 2 1 3 2 4 3 5	-1

input	output
3 2 2 2 0 0 1 3 2 4 3 5	5

input	output
3 2 3 2 1 2 1 3 2 4 3 5	0

Note

In the first sample case, coloring the trees with colors $2, 1, 1$ minimizes the amount of paint used, which equals to $2 + 3 + 5 = 10$. Note that $1, 1, 1$

would not be valid because the beauty of such coloring equals to 1 ($\{1, 1, 1\}$ is a way to group the trees into a single group of the same color).

In the second sample case, all the trees are colored, but the beauty of the coloring is 3, so there is no valid coloring, and the answer is -1.

In the last sample case, all the trees are colored and the beauty of the coloring matches k, so no paint is used and the answer is 0.

D. Bad Luck Island

time limit per test: 2 seconds

memory limit per test: 256 megabytes

The Bad Luck Island is inhabited by three kinds of species: r rocks, s scissors and p papers. At some moments of time two random individuals meet (all pairs of individuals can meet equiprobably), and if they belong to different species, then one individual kills the other one: a rock kills scissors, scissors kill paper, and paper kills a rock. Your task is to determine for each species what is the probability that this species will be the only one to inhabit this island after a long enough period of time.

Input

The single line contains three integers r, s and p $(1 \le r, s, p \le 100)$ — the original number of individuals in the species of rock, scissors and paper, respectively.

Output

Print three space-separated real numbers: the probabilities, at which the rocks, the scissors and the paper will be the only surviving species, respectively. The answer will be considered correct if the relative or absolute error of each number doesn't exceed 10^{-9}.

Examples

input	output
2 2 2	0.333333333333 0.333333333333 0.333333333333

input	output
2 1 2	0.150000000000 0.300000000000 0.550000000000

input	output
1 1 3	0.057142857143 0.657142857143 0.285714285714

E. Pashmak and Graph

time limit per test: 1 second

memory limit per test: 256 megabytes

Pashmak's homework is a problem about graphs. Although he always tries to do his homework completely, he can't solve this problem. As you know, he's really weak at graph theory; so try to help him in solving the problem.

You are given a weighted directed graph with n vertices and m edges. You need to find a path (perhaps, non-simple) with maximum number of edges, such that the weights of the edges increase along the path. In other words, each edge of the path must have strictly greater weight than the previous edge in the path.

Help Pashmak, print the number of edges in the required path.

Input

The first line contains two integers n, m $(2 \le n \le 3 \cdot 10^5;\ 1 \le m \le min(n \cdot (n - 1), 3 \cdot 10^5))$. Then, m lines follows. The i-th line contains three space separated integers: u_i, v_i, w_i $(1 \le u_i, v_i \le n;\ 1 \le w_i \le 10^5)$ which indicates that there's a directed edge with weight w_i from vertex u_i to vertex v_i.

It's guaranteed that the graph doesn't contain self-loops and multiple edges.

Output

Print a single integer — the answer to the problem.

Examples

input

```
3 3
1 2 1
2 3 1
3 1 1
```

output

```
1
```

input

```
3 3
1 2 1
2 3 2
3 1 3
```

output

```
3
```

input

```
6 7
1 2 1
3 2 5
2 4 2
2 5 2
2 6 9
5 4 3
4 3 4
```

output

```
6
```

Note

In the first sample the maximum trail can be any of this trails: $1 \to 2, 2 \to 3, 3 \to 1$.

In the second sample the maximum trail is $1 \to 2 \to 3 \to 1$.

In the third sample the maximum trail is $1 \to 2 \to 5 \to 4 \to 3 \to 2 \to 6$.

F. Little Pony and Harmony Chest

time limit per test: 4 seconds
memory limit per test: 256 megabytes

Princess Twilight went to Celestia and Luna's old castle to research the chest from the Elements of Harmony.

A sequence of positive integers b_i is harmony if and only if for every two elements of the sequence their greatest common divisor equals 1. According to an ancient book, the key of the chest is a harmony sequence b_i which minimizes the following expression:

$$\sum_{i=1}^{n} |a_i - b_i|.$$

You are given sequence a_i, help Princess Twilight to find the key.

Input

The first line contains an integer n $(1 \le n \le 100)$ — the number of elements of the sequences a and b. The next line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 30)$.

Output

Output the key — sequence b_i that minimizes the sum described above. If there are multiple optimal sequences, you can output any of them.

Examples

input
```
5
1 1 1 1 1
```
output
```
1 1 1 1 1
```

input
```
5
1 6 4 2 8
```
output
```
1 5 3 1 8
```

G Spinning up Palindromes

"Sabotage!", exclaimed J. R. Diddly, president and founder of Diddly Widgets Inc.

"Vandalism, perhaps. Nothing's actually been damaged." responded Robert Lackey, the chief accountant.

Both were staring up at the large counter suspended above the factory floor, a counter that had faithfully recorded the number of widgets that had come off the assembly line since the factory was opened. But someone had changed the number being displayed so that it formed...

"It's a palindrome." said Lackey. "It reads the same forwards as backwards."

"What I don't understand," said Diddly, "is why our security guards didn't catch the vandals during their regular sweeps. It must have taken them hours to click forward to this new number, one step at a time."

"No." replied Lackey. "Although we only advance the rightmost digit each time a new widget is built, it's possible to spin any of the digits. With a little planning, this might have taken only a few seconds."

Consider a digital counter consisting of k wheels, each showing a digit from `0` to `9`. Each wheel is mounted so that it can advance to the next digit in a single step, *e.g.*, from `3` to `4`, or from `8` to `9`.

It is also possible to advance from digit `9` to digit `0`. However, when this happens, the wheel on its immediate left will also advance to the next digit automatically. This can have a cascade effect on multiple wheels to the left, but they all happen in a single step.

Given the current setting of the counter, find the smallest number of steps until one can reach a palindrome. The palindrome must respect leading zeros, *e.g.*, `0011` is not a palindrome.

For example, for input `610`, it takes four steps. This can be done by incrementing the `6` wheel four times, resulting in `010`.

Input

The first line of input contains a string of k digits ($1 \leq k \leq 40$), representing the current setting of the counter.

Note that the input may contain leading zeros.

Output

Print, on a single line, the minimum number of wheel advances necessary to produce a palindrome.

Sample Input and Output

0	0
009990001	3
29998	5
610	4
981	2
9084194700940903797191718247801197019268	54