8VC Venture Cup 2017 - Elimination Round

A. PolandBall and Hypothesis

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

PolandBall is a young, clever Ball. He is interested in prime numbers. He has stated a following hypothesis: "`There exists such a positive integer` n `that for each positive integer` m `number` $n \cdot m + 1$ `is a prime number`".

Unfortunately, PolandBall is not experienced yet and doesn't know that his hypothesis is incorrect. Could you prove it wrong? Write a program that finds a counterexample for any n.

Input

The only number in the input is n ($1 \le n \le 1000$) — number from the PolandBall's hypothesis.

Output

Output such m that $n \cdot m + 1$ is not a prime number. Your answer will be considered correct if you output any suitable m such that $1 \le m \le 10^3$. It is guaranteed the the answer exists.

Examples

input
```
3
```
output
```
1
```

input
```
4
```
output
```
2
```

Note

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

For the first sample testcase, $3 \cdot 1 + 1 = 4$. We can output 1.

In the second sample testcase, $4 \cdot 1 + 1 = 5$. We cannot output 1 because 5 is prime. However, $m = 2$ is okay since $4 \cdot 2 + 1 = 9$, which is not a prime number.

B. PolandBall and Game

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

PolandBall is playing a game with EnemyBall. The rules are simple. Players have to say words in turns. You cannot say a word which was already said. PolandBall starts. The Ball which can't say a new word loses.

You're given two lists of words familiar to PolandBall and EnemyBall. Can you determine who wins the game, if both play optimally?

Input

The first input line contains two integers n and m $(1 \le n, m \le 10^3)$ — number of words PolandBall and EnemyBall know, respectively.

Then n strings follow, one per line — words familiar to PolandBall.

Then m strings follow, one per line — words familiar to EnemyBall.

Note that one Ball **cannot** know a word more than once (strings are unique), but some words **can** be known by both players.

Each word is non-empty and consists of no more than 500 lowercase English alphabet letters.

Output

In a single line of print the answer — "YES" if PolandBall wins and "NO" otherwise. Both Balls play optimally.

Examples

input

```
5 1
polandball
is
a
cool
character
nope
```

output

```
YES
```

input

```
2 2
kremowka
wadowicka
kremowka
wiedenska
```

output

```
YES
```

input

```
1 2
a
a
b
```

output

```
NO
```

Note

In the first example PolandBall knows much more words and wins effortlessly.

In the second example if PolandBall says `kremowka` first, then EnemyBall cannot use that word anymore. EnemyBall can only say `wiedenska`. PolandBall says `wadowicka` and wins.

C. PolandBall and Forest

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

PolandBall lives in a forest with his family. There are some trees in the forest. Trees are undirected acyclic graphs with k vertices and $k - 1$ edges, where k is some integer. Note that one vertex **is** a valid tree.

There is exactly one relative living in each vertex of each tree, they have unique ids from 1 to n. For each Ball i we know the id of its most distant relative living on the same tree. If there are several such vertices, we only know the value of the one with smallest id among those.

How many trees are there in the forest?

Input

The first line contains single integer n $(1 \le n \le 10^4)$ — the number of Balls living in the forest.

The second line contains a sequence $p_1, p_2, ..., p_n$ of length n, where $(1 \le p_i \le n)$ holds and p_i denotes the most distant from Ball i relative living on the same tree. If there are several most distant relatives living on the same tree, p_i is the id of one with the smallest id.

It's guaranteed that the sequence p corresponds to some valid forest.

Hacking: To hack someone, you should provide a **correct forest** as a test. The sequence p will be calculated according to the forest and given to the solution you try to hack as input. Use the following format:

In the first line, output the integer n $(1 \le n \le 10^4)$ — the number of Balls and the integer m $(0 \le m < n)$ — the total number of edges in the forest. Then m lines should follow. The i-th of them should contain two integers a_i and b_i and represent an edge between vertices in which relatives a_i and b_i live. For example, the first sample is written as follows:

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

Output

You should output the number of trees in the forest where PolandBall lives.

Interaction

From the technical side, this problem is interactive. However, it should not affect you (except hacking) since there is no interaction.

Examples

input

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

output

```
2
```

input

```
1
1
```

output

```
1
```

Note

In the first sample testcase, possible forest is: `1-2 3-4-5`.

There are 2 trees overall.

In the second sample testcase, the only possible graph is one vertex and no edges. Therefore, there is only one tree.

D. PolandBall and Polygon

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

PolandBall has such a convex polygon with n veritces that no three of its diagonals intersect at the same point. PolandBall decided to improve it and draw some red segments.

He chose a number k such that $gcd(n, k) = 1$. Vertices of the polygon are numbered from 1 to n in a clockwise way. PolandBall repeats the following process n times, starting from the vertex 1:

`Assume you've ended last operation in vertex` x `(consider` $x = 1$ `if it is the first operation). Draw a new segment from vertex` x `to` k`-th next vertex in clockwise direction. This is a vertex` $x + k$ `or` $x + k - n$ `depending on which of these is a valid index of polygon's vertex.`

Your task is to calculate number of polygon's sections after each drawing. A section is a clear area inside the polygon bounded with drawn diagonals or the polygon's sides.

Input

There are only two numbers in the input: n and k ($5 \le n \le 10^6$, $2 \le k \le n - 2$, $gcd(n, k) = 1$).

Output

You should print n values separated by spaces. The i-th value should represent number of polygon's sections after drawing first i lines.

Examples

input
```
5 2
```
output
```
2 3 5 8 11
```

input
```
10 3
```
output
```
2 3 4 6 9 12 16 21 26 31
```

Note

The greatest common divisor (gcd) of two integers a and b is the largest positive integer that divides both a and b without a remainder.

For the first sample testcase, you should output "`2 3 5 8 11`". Pictures below correspond to situations after drawing lines.

E. PolandBall and White-Red graph

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

PolandBall has an undirected simple graph consisting of n vertices. Unfortunately, it has no edges. The graph is very sad because of that. PolandBall wanted to make it happier, adding some red edges. Then, he will add white edges in every remaining place. Therefore, the final graph will be a clique in two colors: white and red.

Colorfulness of the graph is a value $min(d_r, d_w)$, where d_r is the diameter of the red subgraph and d_w is the diameter of white subgraph. The diameter of a graph is a largest value d such that shortest path between some pair of vertices in it is equal to d. If the graph is not connected, we consider its diameter to be `-1`.

PolandBall wants the final graph to be as neat as possible. He wants the final colorfulness to be equal to k. Can you help him and find any graph which satisfies PolandBall's requests?

Input

The only one input line contains two integers n and k ($2 \le n \le 1000$, $1 \le k \le 1000$), representing graph's size and sought colorfulness.

Output

If it's impossible to find a suitable graph, print `-1`.

Otherwise, you can output any graph which fulfills PolandBall's requirements. First, output m — the number of red edges in your graph. Then, you should output m lines, each containing two integers a_i and b_i, ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) which means that there is an undirected red edge between vertices a_i and b_i. Every red edge should be printed exactly once, you can print the edges and the vertices of every edge in arbitrary order.

Remember that PolandBall's graph should remain simple, so no loops or multiple edges are allowed.

Examples

input
```
4 1
```
output
```
-1
```

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

Note

In the first sample case, no graph can fulfill PolandBall's requirements.

In the second sample case, red graph is a path from 1 to 5. Its diameter is 4. However, white graph has diameter 2, because it consists of edges `1-3`, `1-4`, `1-5`, `2-4`, `2-5`, `3-5`.

F. PolandBall and Gifts

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It's Christmas time! PolandBall and his friends will be giving themselves gifts. There are n Balls overall. Each Ball has someone for whom he should bring a present according to some permutation p, $p_i \neq i$ for all i.

Unfortunately, Balls are quite clumsy. We know earlier that exactly k of them will forget to bring their gift. A Ball number i will get his present if the following two constraints will hold:

1. Ball number i will bring the present he should give.
2. Ball x such that $p_x = i$ will bring his present.

What is minimum and maximum possible number of kids who will **not** get their present if exactly k Balls will forget theirs?

Input

The first line of input contains two integers n and k ($2 \le n \le 10^6$, $0 \le k \le n$), representing the number of Balls and the number of Balls who will forget to bring their presents.

The second line contains the permutation p of integers from 1 to n, where p_i is the index of Ball who should get a gift from the i-th Ball. For all i, $p_i \neq i$ holds.

Output

You should output two values — minimum and maximum possible number of Balls who will **not** get their presents, in that order.

Examples

input

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

output

```
2 4
```

input

```
10 1
2 3 4 5 6 7 8 9 10 1
```

output

```
2 2
```

Note

In the first sample, if the third and the first balls will forget to bring their presents, they will be th only balls not getting a present. Thus the minimum answer is 2. However, if the first ans the second balls will forget to bring their presents, then only the fifth ball will get a present. So, the maximum answer is 4.

G. PolandBall and Many Other Balls

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

PolandBall is standing in a row with Many Other Balls. More precisely, there are exactly n Balls. Balls are proud of their home land — and they want to prove that it's strong.

The Balls decided to start with selecting exactly m groups of Balls, each consisting either of single Ball or two neighboring Balls. Each Ball can join no more than one group.

The Balls really want to impress their Enemies. They kindly asked you to calculate number of such divisions for all m where $1 \le m \le k$. Output all these values modulo 998244353, the Enemies will be impressed anyway.

Input

There are exactly two numbers n and k ($1 \le n \le 10^9$, $1 \le k < 2^{15}$), denoting the number of Balls and the maximim number of groups, respectively.

Output

You should output a sequence of k values. The i-th of them should represent the sought number of divisions into exactly i groups, according to PolandBall's rules.

Examples

input
```
3 3
```
output
```
5 5 1
```

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

input
```
5 10
```
output
```
9 25 25 9 1 0 0 0 0 0
```

Note

In the first sample case we can divide Balls into groups as follows:

$\{1\}$, $\{2\}$, $\{3\}$, $\{12\}$, $\{23\}$.

$\{12\}\{3\}$, $\{1\}\{23\}$, $\{1\}\{2\}$, $\{1\}\{3\}$, $\{2\}\{3\}$.

$\{1\}\{2\}\{3\}$.

Therefore, output is: `5 5 1`.