VK Cup 2017 - Квалификация 1

A. Year of University Entrance

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

There is the faculty of Computer Science in Berland. In the social net "TheContact!" for each course of this faculty there is the special group whose name equals the year of university entrance of corresponding course of students at the university.

Each of students joins the group of his course and joins all groups for which the year of student's university entrance differs by no more than x from the year of university entrance of this student, where x — some non-negative integer. A value x is not given, but it can be uniquely determined from the available data. Note that students don't join other groups.

You are given the list of groups which the student Igor joined. According to this information you need to determine the year of Igor's university entrance.

Input

The first line contains the positive odd integer n $(1 \le n \le 5)$ — the number of groups which Igor joined.

The next line contains n distinct integers $a_1, a_2, ..., a_n$ $(2010 \le a_i \le 2100)$ — years of student's university entrance for each group in which Igor is the member.

It is guaranteed that the input data is correct and the answer always exists. Groups are given randomly.

Output

Print the year of Igor's university entrance.

Examples

input
```
3
2014 2016 2015
```
output
```
2015
```

input
```
1
2050
```
output
```
2050
```

Note

In the first test the value $x = 1$. Igor entered the university in 2015. So he joined groups members of which are students who entered the university in 2014, 2015 and 2016.

In the second test the value $x = 0$. Igor entered only the group which corresponds to the year of his university entrance.

B. News About Credit

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

Polycarp studies at the university in the group which consists of n students (including himself). All they are registrated in the social net "TheContacnt!".

Not all students are equally sociable. About each student you know the value a_i — the maximum number of messages which the i-th student is agree to send per day. The student can't send messages to himself.

In early morning Polycarp knew important news that the programming credit will be tomorrow. For this reason it is necessary to urgently inform all groupmates about this news using private messages.

Your task is to make a plan of using private messages, so that:

- the student i sends no more than a_i messages (for all i from 1 to n);
- all students knew the news about the credit (initially only Polycarp knew it);
- the student can inform the other student only if he knows it himself.

Let's consider that all students are numerated by distinct numbers from 1 to n, and Polycarp **always** has the number 1.

In that task you shouldn't minimize the number of messages, the moment of time, when all knew about credit or some other parameters. Find any way how to use private messages which satisfies requirements above.

Input

The first line contains the positive integer n ($2 \leq n \leq 100$) — the number of students.

The second line contains the sequence $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 100$), where a_i equals to the maximum number of messages which can the i-th student agree to send. Consider that Polycarp **always** has the number 1.

Output

Print `-1` to the first line if it is impossible to inform all students about credit.

Otherwise, in the first line print the integer k — the number of messages which will be sent. In each of the next k lines print two **distinct** integers f and t, meaning that the student number f sent the message with news to the student number t. All messages should be printed in chronological order. It means that the student, who is sending the message, must already know this news. It is assumed that students can receive repeated messages with news of the credit.

If there are several answers, it is acceptable to print any of them.

Examples

input

```
4
1 2 1 0
```

output

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

input

```
6
2 0 1 3 2 0
```

output

```
6
1 3
3 4
1 2
4 5
5 6
4 6
```

input

```
3
0 2 2
```

output

```
-1
```

Note

In the first test Polycarp (the student number 1) can send the message to the student number 2, who after that can send the message to students number 3 and 4. Thus, all students knew about the credit.

C. Cycle In Maze

time limit per test: 15 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Robot is in a rectangular maze of size $n \times m$. Each cell of the maze is either empty or occupied by an obstacle. The Robot can move between neighboring cells on the side left (the symbol "L"), right (the symbol "R"), up (the symbol "U") or down (the symbol "D"). The Robot can move to the cell only if it is empty. Initially, the Robot is in the empty cell.

Your task is to find **lexicographically minimal** Robot's cycle with length **exactly** k, which begins and ends in the cell where the Robot was initially. It is allowed to the Robot to visit any cell many times (including starting).

Consider that Robot's way is given as a line which consists of symbols "L", "R", "U" and "D". For example, if firstly the Robot goes down, then left, then right and up, it means that his way is written as "DLRU".

In this task you **don't need** to minimize the length of the way. Find the minimum lexicographical (in alphabet order as in the dictionary) line which satisfies requirements above.

Input

The first line contains three integers n, m and k ($1 \le n, m \le 1000$, $1 \le k \le 10^6$) — the size of the maze and the length of the cycle.

Each of the following n lines contains m symbols — the description of the maze. If the symbol equals to "." the current cell is empty. If the symbol equals to "*" the current cell is occupied by an obstacle. If the symbol equals to "X" then initially the Robot is in this cell and it is empty. It is guaranteed that the symbol "X" is found in the maze exactly once.

Output

Print the lexicographically minimum Robot's way with the length exactly k, which starts and ends in the cell where initially Robot is. If there is no such way, print "IMPOSSIBLE"(without quotes).

Examples

input

```
2 3 2
.**
X..
```

output

```
RL
```

input

```
5 6 14
..***.
*...X.
..*...
..*.**
....*.
```

output

```
DLDDLLLRRRUURU
```

input

```
3 3 4
***
*X*
***
```

output

```
IMPOSSIBLE
```

Note

In the first sample two cyclic ways for the Robot with the length 2 exist — "UD" and "RL". The second cycle is lexicographically less.

In the second sample the Robot should move in the following way: down, left, down, down, left, left, left, right, right, right, up, up, right, up.

In the third sample the Robot can't move to the neighboring cells, because they are occupied by obstacles.

D. k-Interesting Pairs Of Integers

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

Vasya has the sequence consisting of n integers. Vasya consider the pair of integers x and y *k-interesting*, if their binary representation differs from each other exactly in k bits. For example, if $k = 2$, the pair of integers $x = 5$ and $y = 3$ is *k-interesting*, because their binary representation x=`101` and y=`011` differs exactly in two bits.

Vasya wants to know how many pairs of indexes (i, j) are in his sequence so that $i < j$ and the pair of integers a_i and a_j is *k-interesting*. Your task is to help Vasya and determine this number.

Input

The first line contains two integers n and k $(2 \le n \le 10^5, 0 \le k \le 14)$ — the number of integers in Vasya's sequence and the number of bits in which integers in *k-interesting* pair should differ.

The second line contains the sequence $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^4)$, which Vasya has.

Output

Print the number of pairs (i, j) so that $i < j$ and the pair of integers a_i and a_j is *k-interesting*.

Examples

input

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

output

```
4
```

input

```
6 0
200 100 100 100 200 200
```

output

```
6
```

Note

In the first test there are 4 *k-interesting* pairs:

- $(1, 3)$,
- $(1, 4)$,
- $(2, 3)$,
- $(2, 4)$.

In the second test $k = 0$. Consequently, integers in any *k-interesting* pair should be equal to themselves. Thus, for the second test there are 6 *k-interesting* pairs:

- $(1, 5)$,
- $(1, 6)$,
- $(2, 3)$,
- $(2, 4)$,
- $(3, 4)$,
- $(5, 6)$.

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform