

Divide

You are given sequence of N integer numbers $A[0], A[1], \dots, A[N - 1]$ and integer number M . What is the maximal number of non-empty parts you can split the sequence to, so that XOR (excluding OR) of all numbers in each part is not exceeding M . Each number of sequence must be in exactly one part. Each part must be a continuous subsequence.

Examples

In the first example $N = 4, M = 4$.

	0	1	2	3
A	1	2	3	4

In this example maximal number of parts is 4, XOR of each part doesn't exceed M , so this is an optimal solution.

In the second example $N = 4, M = 2$.

	0	1	2	3
A	1	2	3	4

In this example it's not possible to split the sequence according to the conditions so the answer is -1 .

In the third example $N = 4, M = 3$.

	0	1	2	3
A	1	5	6	2

In this example optimal solution consists of following parts: (1), (5, 6) and (2).

Task

Given the sequence of numbers, find the maximum number of parts you can split the sequence to or determine that it is impossible.

- `solve(N, M, A)`;
 - N : number of numbers in sequence A . $1 \leq N \leq 10^5$.
 - M : value that each part must not exceed. $0 \leq M \leq 10^9$.

- A : array of integers. $0 \leq A[i] \leq 10^9$, $0 \leq i < N$.
- The function should return maximal number of parts, or -1 if he can not split the sequence.

Subtasks

subtask	points	N
1	20	$1 \leq N \leq 10$
2	30	$1 \leq N \leq 1,000$
3	50	$1 \leq N \leq 100,000$

Implementation details

You have to submit exactly one file, called `divide.c`, `divide.cpp`, `divide.pas` or `divide.java`. This file should implement the subprogram described above, using the following signatures.

C/C++ program (include `divide.h` at the top of the source file)

```
int solve(int N, int M, int A[]);
```

Pascal programs (implement the described method in unit `divide`)

```
function solve(N,M :longint; var A: array of longint) : longint;
```

Java programs (implement the described method in public class `divide`)

```
public int solve(int N, int M, int[] A);
```

Sample grader

The sample grader reads the input in the following format:

- line 1: $N M$
- line 2: $A[0], \dots, A[N-1]$

The sample grader will print the return value of `divide`.