Problem C.

Input file:	`c.in`
Output file:	`c.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given numbers N, x and a sequence of N numbers. Find the largest possible interval of consequently following elements, such that "xor"of these elements is not less than x. I.e., more formally, find such i and k that

$a_i \oplus a_{i+1} \oplus \cdots \oplus a_{i+k-1} \geq x$, $1 \leq i \leq i+k-1 \leq N$,

and k is largest possible positive number.

It's guaranteed that for each test from the testset such an interval exists.

We remind you that $xor(\oplus)$ operation is applied to numbers in binary representation, so that for each pair of bits the following is true:

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

The result of this operation doesn't depend on the order of operands $a \oplus b = b \oplus a$. Moreover $(a \oplus (a \oplus b)) = b$.

In Pascal this operation is represented as xor. In C/C++/Java as $\wedge$.

Input

The first line of input contains N $(1 \leq N \leq 250\,000)$ and x $(0 \leq x \leq 1\,000\,000\,000)$. The second line of input contains N non-negative numbers not exceeding 10^9.

Output

The first line of output must contain two numbers: i and k. In case of many solutions output the one with the smallest i.

Examples

c.in	c.out

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

Note

In 40% testcases $N \leq 35\,000$.

In 80% testcases $N \leq 100\,000$.