Given an array, your goal is to find, for each element, the largest subarray containing it whose cost is at least k.

Specifically, let $A = [A_1, A_2, \ldots, A_n]$ be an array of length n, and let $A_{l\ldots r} = [A_l, \ldots, A_r]$ be the subarray from index l to index r. Also,

- Let $\text{MAX}(l, r)$ be the largest number in $A_{l\ldots r}$.
- Let $\text{MIN}(l, r)$ be the smallest number in $A_{l\ldots r}$.
- Let $\text{OR}(l, r)$ be the bitwise OR of the elements of $A_{l\ldots r}$.
- Let $\text{AND}(l, r)$ be the bitwise AND of the elements of $A_{l\ldots r}$.

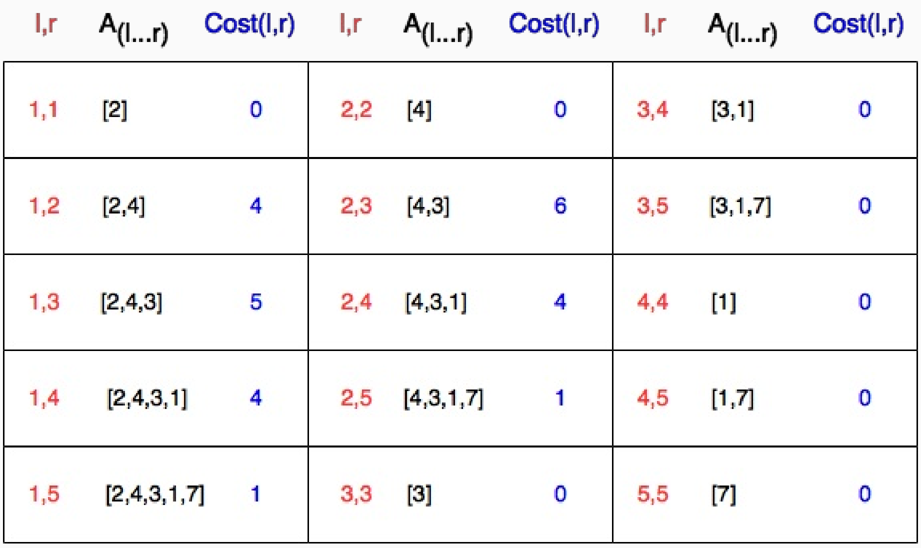
The *cost* of $A_{l\ldots r}$, denoted $cost(l, r)$, is defined as

$$cost(l, r) = (\text{OR}(l, r) - \text{AND}(l, r)) - (\text{MAX}(l, r) - \text{MIN}(l, r)).$$

The *size* of $A_{l\ldots r}$ is defined as $r - l + 1$.

You are given the array A and and an integer k. For each index i from 1 to n, your goal is to find the largest size of any subarray $A_{l\ldots r}$ such that $1 \leq l \leq i \leq r \leq n$ and $cost(l, r) \geq k$.

Consider, array $A = [2, 4, 3, 1, 7]$ and $k = 6$. The possible sub-arrays and their costs would be as follows:



l,r	A(l...r)	Cost(l,r)	l,r	A(l...r)	Cost(l,r)	l,r	A(l...r)	Cost(l,r)
1,1	[2]	0	2,2	[4]	0	3,4	[3,1]	0
1,2	[2,4]	4	2,3	[4,3]	6	3,5	[3,1,7]	0
1,3	[2,4,3]	5	2,4	[4,3,1]	4	4,4	[1]	0
1,4	[2,4,3,1]	4	2,5	[4,3,1,7]	1	4,5	[1,7]	0
1,5	[2,4,3,1,7]	1	3,3	[3]	0	5,5	[7]	0

Complete the function `costlyIntervals` which takes two integers n and k as first line of input, and array $A_1, A_2, \ldots, A_n$ in the second line of input. Return an array of n integers, where the i^{th} element contains the answer for index i of the input array, $1 \leq i \leq n$. Every element of the output array denotes the largest size of a subarray containing i whose cost is at least k, or -1 if there is no such subarray.

Constraints

- $1 \leq n \leq 10^5$
- $0 \leq A_i \leq 10^9$
- $0 \leq k \leq 10^9$

Subtasks

- For 5% of the maximum score, $n \leq 100$.
- For 15% of the maximum score, $n \leq 5 \cdot 10^3$.

Sample Input

$n = 5$, $k = 6$
$A = [2, 4, 3, 1, 7]$

Sample Output

$[-1, 2, 2, -1, -1]$

Explanation

In this example, we have $k = 6$. There is only one subarray whose cost is at least 6, and that is $A_{2...3} = [4, 3]$, since $cost(2, 3) = 6$. Its size is 2. Thus, for $i = 2$ and $i = 3$, the answer is 2, and for the others, -1.