

## Problem L. Not Our Problem

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Someone (not me!) calls an array of length  $n$  consisting of non-negative integers *good* if and only if  $a_i \cdot a_{i+1} \cdot \min(a_i, a_{i+1}) \leq C$  holds for all  $1 \leq i < n$ .

Someone else (we have nothing to do with them!) gives you an array of length  $n$  with some blank positions. Calculate the number of ways to fill in the blank positions so that the array is good, or determine that there are infinitely many ways. If the answer is finite, print it modulo 998 244 353.

### Input

The first line contains two integers  $n$  and  $C$  ( $1 \leq n \leq 10^6$ ,  $0 \leq C \leq 10^{18}$ ) — the length of the array and the constraint on the product.

The second line contains the array  $a$ . Each element is either  $-1$ , which means that it needs to be filled, or between  $0$  and  $10^{18}$ , inclusive.

### Output

If the number of ways is infinite, print  $-1$ , otherwise print the number of ways modulo 998 244 353.

### Examples

standard input	standard output
4 100 -1 7 2 -1	104
4 100 10 -1 -1 1	240
1 0 -1	-1
2 10 10 10	0