Sequence Folding

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

Given is a binary sequence $a_1, a_2, \ldots, a_n$, where n is a power of two. In one move, you can choose any of its elements and change it to its opposite (i.e., choose index i from the interval $[1, n]$ and change a_i to $1 - a_i$). Additionally, if at some point the sequence is a palindrome (i.e., for every valid index i, $a_i = a_{n+1-i}$ holds), you can cut the right half of the sequence, leaving only $a_1, a_2, \ldots, a_{\frac{n}{2}}$, and then change n to the length of the new sequence (i.e., divide it by 2). Such an action is not counted as a move – by the number of moves, we mean only the number of element changes. You can freely interleave changes of the sequence elements and halving its length. Of course, you can't shorten the sequence if it consists of only one element.

Determine the minimum number of moves needed to make the sequence consist of only one element.

Additionally, n can be large, and the sequence is given by a list of positions of ones in it (the rest of the elements are zeros).

Input

In the first line of standard input, there are two integers n and m ($1 \leq n \leq 10^{18}$; $1 \leq m \leq 100\,000$; $m \leq n$; n is a power of 2 with a non-negative integer exponent), representing respectively the length of the sequence and the number of ones in it.

In the second line, there is a sequence of m integers $p_1, p_2, \ldots, p_m$ ($1 \leq p_i \leq n$; $p_i < p_{i+1}$), indicating the positions of the consecutive ones in the sequence $a_1, a_2, \ldots, a_n$.

Output

The output should contain one integer, indicating the minimum number of moves required to shorten the sequence $a_1, a_2, \ldots, a_n$ to a single-element sequence.

Example

standard input	standard output
8 3 1 5 8	2

Note

In the sample test, the sequence `10001001` is given. There are three ways to shorten it to one element using two moves:

- `10001001` $\xrightarrow[+1]{}$ `10011001` $\longrightarrow$ `1001` $\longrightarrow$ `10` $\xrightarrow[+1]{}$ `11` $\longrightarrow$ `1`
- `10001001` $\xrightarrow[+1]{}$ `10011001` $\longrightarrow$ `1001` $\longrightarrow$ `10` $\xrightarrow[+1]{}$ `00` $\longrightarrow$ `0`
- `10001001` $\xrightarrow[+1]{}$ `10000001` $\longrightarrow$ `1000` $\xrightarrow[+1]{}$ `0000` $\longrightarrow$ `00` $\longrightarrow$ `0`

There is no way to shorten the sequence to one element in fewer than two moves, so the result is the number 2.