

Necklace

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 512 megabytes

Bob has given Alice a necklace as her birthday gift. That necklace has N crystals, M of which catches her fancy. Formally, crystals with labels of $1..N$ are strung sequentially in a circle. And those catching her fancy are labelled n_1, n_2, \dots, n_M .

Alice would like the necklace to be divided into M pieces. She asked Bob to sever the necklace into M pieces. Each piece should consist of a sequence of crystals as in the original necklace (i.e., the crystal labels are contiguous, except that 1 following N is accepted) and should include one crystal catching Alice's fancy. Obviously, each crystal must belong to exactly one piece.

However, an excessively long piece of severed necklaces could not fit Alice's slim neck. So she wants to know, among all possible severings as requested, what the minimum length of the longest piece from a severed necklace is. She wants you to answer this question.

Input

The first line of the input data contains 2 integers N and M ($1 \leq M \leq N < 10^{18}$ and $M \leq 10^6$).

The second line contains M integers, the i -th of which represents n_i . The n_i 's are given in ascending order.

Output

You need to output your answer in one line.

Example

| standard input | standard output |
|-----------------|-----------------|
| 10 4 2 5 6 8 | 3 |

Note

As for the example: You can sever the necklace into $[1, 3]$, $[4, 5]$, $[6, 7]$, $[8, 10]$.

Considering huge scale of data, here is a way provided to help input faster:

```
#define gc() (is==it?it=(is=in)+fread(in,1,Q,stdin),(is==it?EOF:*is++):*is++)
const int Q=(1<<24)+1;
char in[Q],*is=in,*it=in,c;
void read(long long &n){
    for(n=0;(c=gc())<'0' || c>'9';);
    for(;c<='9'&& c>='0';c=gc())n=n*10+c-48;
}
```