

E. Candy Shop

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The prestigious Codeforces kindergarten consists of n kids, numbered 1 through n . Each of them are given allowance in rubles by their parents.

Today, they are going to the most famous candy shop in the town. The shop sells candies in packages: for all i between 1 and m , inclusive, it sells a package containing exactly i candies. A candy costs one ruble, so a package containing x candies costs x rubles.

The kids will purchase candies in turns, starting from kid 1. In a single turn, kid i will purchase one candy package. Due to the highly competitive nature of Codeforces kindergarten, during a turn, the number of candies contained in the package purchased by the kid will always be strictly greater than the number of candies contained in the package purchased by the kid in the preceding turn (an exception is in the first turn: the first kid may purchase any package). Then, the turn proceeds to kid $i + 1$, or to kid 1 if it was kid n 's turn. This process can be ended at any time, but at the end of the purchase process, all the kids must have the same number of candy packages. Of course, the amount spent by each kid on the candies cannot exceed their allowance.

You work at the candy shop and would like to prepare the candies for the kids. Print the maximum number of candies that can be sold by the candy shop to the kids. If the kids cannot purchase any candy (due to insufficient allowance), print 0.

Input

The first line contains two space-separated integers n and m ($2 \leq n \leq 2 \cdot 10^5$, $2 \leq m \leq 5 \cdot 10^6$, $n \leq m$), denoting the number of kids and the maximum number of candies in a package sold by the candy shop, respectively.

Then n lines follow, each line will contain a single positive integer not exceeding denoting the allowance of a kid in rubles. The allowances are given in order from kid 1 to kid n .

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is recommended to use `cin`, `cout` streams (also you may use `%I64d` specifier).

Output

Print a single integer denoting the maximum number of candies that can be sold by the candy shop.

Examples

input
2 5 5 10
output
13

input
3 8 8 16 13
output
32

input
2 5000000 1250000250000 1250000250000
output
1250000250000

Note

For the first example, one of the scenarios that will result in 13 purchased candies is as follows.

- Turn 1. Kid 1 purchases 1 candy.
- Turn 2. Kid 2 purchases 3 candies.
- Turn 3. Kid 1 purchases 4 candies.
- Turn 4. Kid 2 purchases 5 candies.