Awesome Frog

Input File: *frogin.txt*
Output File: *frogout.txt*

Time Limit: 1 second

The inaugural International Olympiad in Frogleaping is being held in Australia in 2013 and you are determined to win. While you want nothing to do with such slimy, jumpy creatures, you plan to enter a frog-like robot that you know will be faster than all the other organic entrants.

The IOF takes place in a large pond in which there is a sequence of lily pads arranged in a long line. The rules of the race are simple: your frog will be placed on the first lily pad, then it must jump to the second lily pad, then the third and so forth until it reaches the last lily pad in the course. Note that you can not 'skip' lily pads—every lily pad must be jumped on exactly once. The first frog to reach the last lily pad will win the race. Since your robotic frog has super-frog speed, you are confident in your victory.

However, your frog has one minor incorrectable flaw—it is only able to jump one fixed distance. Specifically, it can only jump *exactly* K metres forward from its current location, even if this lands the frog in the water (where it will promptly short-circuit).

Since the initial lily pad positions may make it impossible for your frog to reach the last lily pad, you plan to create a distraction and move the lily pads so that they are spaced exactly K metres apart, enabling your frog to jump from the first to the last without falling in the water. Shifting a lily pad by one metre will take you one second, and the longer you spend stealthily moving lily pads, the more likely that the IOF judges will notice and disqualify you from the competition.

Given the initial distances between the lily pads in the course, you must write a program to compute the minimum time you will have to spend shifting lily pads such that all pairs of consecutive lily pads are exactly K metres apart. You can assume that the pond is sufficiently long so that the first lily pad can be moved any distance back, and the last lily pad can be moved any distance forward.

Input

Your program should read from the file `frogin.txt`. The first line of this file will consist of two space-separated integers N and K. The following $N-1$ lines will contain the initial distances between consecutive pairs of lily pads. Specifically, the ith line will contain one integer representing the distance between the ith and $(i+1)$th lily pad.

Output

Your program should write to the file `frogout.txt`. Your output file should consist of one line containing one integer: the minimum total time spent shifting lily pads so as to ensure your victory.

Sample Input

```
5 6
8
3
6
4
```

Sample Output

```
6
```

Explanation

In this example, if we shift the first lily pad back by 1 metre and the second lily pad back by 3 metres, there will be 6 metres between the first and second and 6 metres between the second and third lily pads. The third and fourth are already 6 metres apart, but we must move the fifth lily pad forward by 2 metres to put it 6 metres after the fourth.

The total time taken is $1 + 3 + 2 = 6$ seconds.

Constraints

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $1 \leq K \leq 20,000$
- $2 \leq N \leq 100,000$
- The initial distance between each consecutive pair of lily pads will be between 1 and $20,000$ metres.
- It is guaranteed that the minimum time you will have to spend shifting lily pads will be at most $2^{31} - 1$.

As some of the test cases will be quite large, you may need to think about how well your solution scales for larger input values. However, not all the cases will be large. In particular:

- For 30% of cases, $N \leq 100$, and distances between consecutive pairs of lily pads will be at most $1,000$ metres.

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.