parking

You have to manage a parking lot with M parking spaces. The parking spaces are labeled with integer numbers starting from *1* to M. All the parking spaces are placed one after one, such as the i^{th} parking space has as neighbors the $(i-1)^{th}$ and $(i+1)^{th}$ parking spaces, except the 1^{st} and the M^{th} parking spaces that only have the 2^{nd} and the $(M-1)^{th}$ parking spaces as neighbors, respectively.

Initially, all the parking spaces are empty. There are N drivers and each of them has their favorite parking space (the i^{th} driver would like to park on the A_i^{th} parking space). If the i^{th} driver ends up parking on the B_i^{th} parking space then we can measure his dissatisfaction by using the following formula: $|A_i - B_i|$.

Task

Given M the number of parking spaces, N the number of drivers and A_i the favorite parking spaces of all the drivers, you are to compute the minimal cumulative dissatisfaction of all the drivers.

Input data

The first line of the input file `parking.in` contains two numbers M and N, separated by one space. The second line contains the favorite parking spaces $A_1, A_2, \ldots, A_n$ of each of the drivers.

Output data

The first line of the output file `parking.out` must contain a single number representing the minimal cumulative dissatisfaction of all the drivers.

Limits and constraints

- $1 \leq M \leq 10{,}000$
- $1 \leq N \leq 1{,}000$
- $N \leq M$
- Two or more drivers cannot use the same parking space.
- The cumulative dissatisfaction of all the drivers is computed by summing the dissatisfactions of each driver.
- Time limit: 0.1 seconds
- Memory limit: 2 MB

Example

parking.in	parking.out	Explanations
10 3 3 5 3	1	One way of achieving the minimal cumulative dissatisfaction of all the drivers is to instruct the drivers to park as follows: $B_1 = 3, B_2 = 5$ and $B_3 = 4$. The dissatisfactions of the drivers are 1, 0 and 0, respectively, so their cumulative dissatisfaction is 1 + 0 + 0 = 1.