

# MINIMUM DISTANCE

## CHALLENGE DESCRIPTION:

Alice is looking for a sorority to join for her first year at Acme University. There is a street consisting entirely of sorority houses near the university, and some of her high school friends have already joined sorority houses on the street. (More than one of her friends may live at the same sorority house.)

Alice wants to visit her friends frequently, and needs a program that will help her pick an optimal house to visit them from. Each sorority house has a street number that indicates its location on the street. The optimal location will minimize the sum of differences between the number of Alice's house and the number of her friends' houses.

For example: Alice's friends live at houses 3, 3, 5, and 7. Alice moves in at house 4. Then the distances to her friends' houses are 1, 1, 1, and 3, totaling 6.

## INPUT SAMPLE:

The input consists of several integers on a line, separated by spaces. The first integer F contains the number of friends ( $0 < F < 100$ ). Then F street addresses A follow ( $0 < A < 10000$ ).

For example:

```
4 3 3 5 7
3 20 30 40
```

## OUTPUT SAMPLE:

Print a line containing the minimal sum of distances for an optimal sorority location.

For example:

```
6
20
```

## CONSTRAINTS:

- 1. Number of friends:  $0 < F < 100$
- 2. Street addresses:  $0 < A < 10000$
- 3. Number of test cases is 10.