

## A. Plus One on the Subset

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Polycarp got an array of integers  $a[1 \dots n]$  as a gift. Now he wants to perform a certain number of operations (possibly zero) so that all elements of the array become the same (that is, to become  $a_1 = a_2 = \dots = a_n$ ).

- In one operation, he can take some indices in the array and increase the elements of the array at those indices by 1.

For example, let  $a = [4, 2, 1, 6, 2]$ . He can perform the following operation: select indices 1, 2, and 4 and increase elements of the array in those indices by 1. As a result, in one operation, he can get a new state of the array  $a = [5, 3, 1, 7, 2]$ .

What is the minimum number of operations it can take so that all elements of the array become equal to each other (that is, to become  $a_1 = a_2 = \dots = a_n$ )?

### Input

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases in the test.

The following are descriptions of the input test cases.

The first line of the description of each test case contains one integer  $n$  ( $1 \leq n \leq 50$ ) — the array size.

The second line of the description of each test case contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — elements of the array.

### Output

For each test case, print one integer — the minimum number of operations to make all elements of the array equal.

### Example

input
3 6 3 4 2 4 1 2 3 1000 1002 998 2 12 11
output
3 4 1

### Note

First test case:

- $a = [3, 4, 2, 4, 1, 2]$  take  $a_3, a_5$  and perform an operation plus one on them, as a result we get  $a = [3, 4, 3, 4, 2, 2]$ .
- $a = [3, 4, 3, 4, 2, 2]$  we take  $a_1, a_5, a_6$  and perform an operation on them plus one, as a result we get  $a = [4, 4, 3, 4, 3, 3]$ .
- $a = [4, 4, 3, 4, 3, 3]$  we take  $a_3, a_5, a_6$  and perform an operation on them plus one, as a result we get  $a = [4, 4, 4, 4, 4, 4]$ .

There are other sequences of 3 operations, after the application of which all elements become equal.

Second test case:

- $a = [1000, 1002, 998]$  2 times we take  $a_1, a_3$  and perform an operation plus one on them, as a result we get  $a = [1002, 1002, 1000]$ .
- $a = [1002, 1002, 1000]$  also take  $a_3$  2 times and perform an operation plus one on it, as a result we get  $a = [1002, 1002, 1002]$ .

Third test case:

- $a = [12, 11]$  take  $a_2$  and perform an operation plus one on it, as a result we get  $a = [12, 12]$ .