

Preprocessor Solution

Preprocessor directives are lines included in the code preceded by a hash sign (#). These lines are directives for the preprocessor. The preprocessor examines the code before actual compilation of code begins and resolves all these directives before any code is actually generated by regular statements.

```
#define INF 10000000
if( val == INF) {
//Do something
}
After the preprocessor has replaced the directives, the code will be
if( val == 10000000) { //Here INF is replaced by the value with which it's defined.
//Do something
}
```

You can also define function macros which have parameters.

```
#define add(a, b) a + b
int x = add(a, b);

The second statement after the preprocessor has replaced the directives will be:
int x = a + b;
```

To know more about preprocessor directives, you can go to this [link](#)

You're spending your afternoon at a local school, teaching kids how to code. You give them a simple task: find the difference between the maximum and minimum values in an array of integers.

After a few hours, they came up with some promising source code. Unfortunately, it doesn't compile! Since you don't want to discourage them, you decide to make their code work without modifying it by adding preprocessor macros.

Review the locked stub code in your editor and add the preprocessor macros necessary to make the code work.

Input Format

The first line contains an integer, N , denoting the size of the array.
The second line contains N space-separated integers, x_1, x_2, \dots, x_n , describing the elements in the array.

Constraints

- $1 \leq N \leq 10^3$
- $-10^8 \leq x_i \leq 10^8$

Output Format

You are not responsible for printing any output to stdout. Once the necessary preprocessor macros are written, the locked code in your editor will print a line that says **Result = Z**, where Z is the difference between the maximum and minimum values in the array.

Sample Input

5
32 332 -23 -154 65

Sample Output

Result = 486

Explanation

$$332 - (-154) = 486$$