A. Difference Row

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You want to arrange n integers $a_1, a_2, ..., a_n$ in some order in a row. Let's define the value of an arrangement as the sum of differences between all pairs of adjacent integers.

More formally, let's denote some arrangement as a sequence of integers $x_1, x_2, ..., x_n$, where sequence x is a permutation of sequence a. The value of such an arrangement is $(x_1 - x_2) + (x_2 - x_3) + ... + (x_{n-1} - x_n)$.

Find the largest possible value of an arrangement. Then, output the lexicographically smallest sequence x that corresponds to an arrangement of the largest possible value.

Input

The first line of the input contains integer n $(2 \le n \le 100)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(|a_i| \le 1000)$.

Output

Print the required sequence $x_1, x_2, ..., x_n$. Sequence x should be the lexicographically smallest permutation of a that corresponds to an arrangement of the largest possible value.

Examples

input

```
5
100 -100 50 0 -50
```

output

```
100 -50 0 50 -100
```

Note

In the sample test case, the value of the output arrangement is $(100 - (-50)) + ((-50) - 0) + (0 - 50) + (50 - (-100)) = 200$. No other arrangement has a larger value, and among all arrangements with the value of 200, the output arrangement is the lexicographically smallest one.

Sequence $x_1, x_2, ... , x_p$ is *lexicographically smaller* than sequence $y_1, y_2, ... , y_p$ if there exists an integer r $(0 \le r < p)$ such that $x_1 = y_1, x_2 = y_2, ... , x_r = y_r$ and $x_{r+1} < y_{r+1}$.

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, then value 800 ms will be displayed and used to determine the verdict.

Codeforces Round #201 (Div. 2)

Finished

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Problem tags

implementation sortings

No tag edit access

→ Contest materials

- Announcement
- Tutorial