Remove outliers
Problem ID: a12p04removeoutliers

Write two functions for removing the minimum number and the maximum number from integer lists:

- `without_outliers(a_list)`:
  Returns a new list. Does not modify the list passed as an argument.
- `remove_min_and_max(a_list)`:
  Returns nothing, modifies the list passed as an argument.

The main file, which handles input and output, is already provided. - Please only submit your function definitions, without any code outside the functions!

Input

Each of the functions should accept one list l as a parameter. You may assume the elements of the list l will be integers.

In the samples below, the input consists of one line containing a sequence $x = (x_1, \ldots, x_n)$ of n numbers separated by a space. Gradescope will handle reading the input, splitting it, converting to integers and passing the resulting list as an argument to the functions.

In the tests, x will always satisfy $2 \leq n \leq 1\,000$ and $0 \leq x_i < 10\,000$ for all i with $1 \leq i \leq n$.

Output

The function `without_outliers()` should return a new list m containing all the numbers appearing in l except two, the largest and the smallest, in the same order as they appear in l, and no other numbers. After the function is called, the original list l should remain unchanged.

The function `remove_min_and_max()` should return nothing (just `None`), but should modify the given list l, so that after the function is called, the list will have changed to a modified list l' containing all the numbers appearing in l except two, the largest and the smallest, in the same order as they appear in l, and no other numbers.

In each test case, Gradescope will call both of the functions with the given list as a parameter, in sequence, and will verify the results as well as the effects on the given list, and print the following four lines to the output, as seen in the samples below:

- "`Original list before calling functions:` {l}"
- "`Resulting list after extracting middle:` {m}"
- "`Original list after extracting middle and before removing outliers:` {l}"
- "`Original list after removing outliers:` {l'}"

Sample Input 1

```
21 1 23 102
```

Sample Output 1

```
Original list before calling functions: [21, 1, 23, 102]
Resulting list after extracting middle: [21, 23]
Original list after extracting middle and before removing outliers: [21, 1, 23, 102]
Original list after removing outliers: [21, 23]
```

Sample Input 2

```
5 4 3 2 1 0
```

Sample Output 2

```
Original list before calling functions: [5, 4, 3, 2, 1, 0]
Resulting list after extracting middle: [4, 3, 2, 1]
Original list after extracting middle and before removing outliers: [5, 4, 3, 2, 1, 0]
Original list after removing outliers: [4, 3, 2, 1]
```

Sample Input 3

```
3 9 5 1 6 8
```

Sample Output 3

```
Original list before calling functions: [3, 9, 5, 1, 6, 8]
Resulting list after extracting middle: [3, 5, 6, 8]
Original list after extracting middle and before removing outliers: [3, 9, 5, 1, 6, 8]
Original list after removing outliers: [3, 5, 6, 8]
```

Sample Input 4

```
9 2 3 6 1 8 7
```

Sample Output 4

```
Original list before calling functions: [9, 2, 3, 6, 1, 8, 7]
Resulting list after extracting middle: [2, 3, 6, 8, 7]
Original list after extracting middle and before removing outliers: [9, 2, 3, 6, 1, 8, 7]
Original list after removing outliers: [2, 3, 6, 8, 7]
```