E. Choosing Carrot

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

Oleg the bank client and Igor the analyst are arguing again. This time, they want to pick a gift as a present for their friend, ZS the coder. After a long thought, they decided that their friend loves to eat carrots the most and thus they want to pick the best carrot as their present.

There are n carrots arranged in a line. The i-th carrot from the left has juiciness a_i. Oleg thinks ZS loves juicy carrots whereas Igor thinks that he hates juicy carrots. Thus, Oleg would like to maximize the juiciness of the carrot they choose while Igor would like to minimize the juiciness of the carrot they choose.

To settle this issue, they decided to play a game again. Oleg and Igor take turns to play the game. In each turn, a player can choose a carrot from either end of the line, and eat it. The game ends when only one carrot remains. Oleg moves first. The last remaining carrot will be the carrot that they will give their friend, ZS.

Oleg is a sneaky bank client. When Igor goes to a restroom, he performs k moves before the start of the game. Each move is the same as above (eat a carrot from either end of the line). After Igor returns, they start the game with Oleg still going first.

Oleg wonders: for each k such that $0 \le k \le n - 1$, what is the juiciness of the carrot they will give to ZS if he makes k extra moves beforehand and both players play optimally?

Input

The first line of input contains a single integer n $(1 \le n \le 3 \cdot 10^5)$ — the total number of carrots.

The next line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$. Here a_i denotes the juiciness of the i-th carrot from the left of the line.

Output

Output n space-separated integers $x_0, x_1, ..., x_{n-1}$. Here, x_i denotes the juiciness of the carrot the friends will present to ZS if $k = i$.

Examples

input

```
4
1 2 3 5
```

output

```
3 3 5 5
```

input

```
5
1000000000 1000000000 1000000000 1000000000 1
```

output

```
1000000000 1000000000 1000000000 1000000000 1000000000
```

Note

For the first example,

When $k = 0$, one possible optimal game is as follows:

- Oleg eats the carrot with juiciness 1.

- Igor eats the carrot with juiciness 5.
- Oleg eats the carrot with juiciness 2.
- The remaining carrot has juiciness 3.

When $k = 1$, one possible optimal play is as follows:

- Oleg eats the carrot with juiciness 1 beforehand.
- Oleg eats the carrot with juiciness 2.
- Igor eats the carrot with juiciness 5.
- The remaining carrot has juiciness 3.

When $k = 2$, one possible optimal play is as follows:

- Oleg eats the carrot with juiciness 1 beforehand.
- Oleg eats the carrot with juiciness 2 beforehand.
- Oleg eats the carrot with juiciness 3.
- The remaining carrot has juiciness 5.

When $k = 3$, one possible optimal play is as follows:

- Oleg eats the carrot with juiciness 1 beforehand.
- Oleg eats the carrot with juiciness 2 beforehand.
- Oleg eats the carrot with juiciness 3 beforehand.
- The remaining carrot has juiciness 5.

Thus, the answer is $3, 3, 5, 5$.

For the second sample, Oleg can always eat the carrot with juiciness 1 since he always moves first. So, the remaining carrot will always have juiciness 1000000000.