COCI '07 Contest 2 #4 Turbo

Time Limit: 1.0s   **Memory Limit:** 32M

Frane has been given the task of sorting an array of numbers. The array consists of N integers, each between 1 and N (inclusive), with each of those appearing exactly once in the array. Frane has come up with the following sorting algorithm which operates in N phases, and named it turbosort:

- In the first phase, the number 1 is moved to position 1 by repeatedly swapping consecutive elements.
- In the second phase, the number N is moved to position N in the same manner.
- In the third phase, the number 2 is moved to position 2.
- In the fourth phase, the number $N - 1$ is moved to position $N - 1$.
- And so on.

In other words, when the number of the phase is odd, Frane will choose the smallest number not yet chosen, and move it to its final position. In even phases he chooses the largest number not yet chosen.

Write a program which, given the initial array, output the number of swaps in each phase of the algorithm.

Input Specification

The first line contains an integer N $(1 \le N \le 100\,000)$, the number of elements in the array. Each of the following N lines contains an integer between 1 and N (inclusive), the array to be sorted. The array will contain no duplicates.

Output Specification

For each of the N phases, output the number of swaps on a single line.

Scoring

In test cases worth 70% of points, N will be less than 100.

Sample Input 1

```
3
2
1
3
```

Sample Output 1

```
1
0
0
```

Sample Input 2

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

Sample Output 2

```
4
3
2
1
0
```

Sample Input 3

```
7
5
4
3
7
1
2
6
```

Sample Output 3

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