Sort the following arrays by hand in smallest to largest order using selection sort and bubble sort.

8	4	6	1	3	5

794	322	284	666	833	534

Here one example of how to do it:

Selection sort

5	4	3	2	1
1	**4**	3	2	5
1	2	**3**	4	5
1	2	3	**4**	5
1	2	3	4	**5**

Unsorted array
Sorted array

Bubble sort

5	**4**	3	2	1
4	**5**	**3**	2	1
4	3	**5**	**2**	1
4	3	2	**5**	**1**
4	**3**	2	1	5
3	**4**	**2**	1	5
3	2	**4**	**1**	5
3	2	1	**4**	**5**
3	**2**	1	4	5
2	**3**	**1**	4	5
2	1	**3**	**4**	5
2	1	3	**4**	**5**
2	**1**	3	4	5
1	2	3	4	5

Why bubble sort and selection sort are $O(n^2)$?

How you already know how the previous algorithms works, it's time to solve a problem with these algorithms

Maria is a teacher who wants to give a prize to the student with the best grade and give to the worst student extra classes, if two or more students get the highest or lowest grade, the teacher will need to buy more prizes or give extra classes to more students, she wants to know how many prices to buy and how many students will be taking extra classes, the inputs are n (number of students) and a_n (grade of

each student), the output are two numbers p (number of prizes) and c (number of students taking extra classes), you should sort using bubble sort or selection sort to solve the problem.

Input

3

10 5 3

Output

1 1

Explanation: since there are just 1 highest grade (10) and 1 lowest grade (3), the output is 1 1

Input

7

10 3 5 10 4 2 4

Output

2 1

Explanation: since there are two highest grades (10) and just one lowest grade (2), the output is 2 1