

Problem: 1

Write a program that takes an array of numbers between 1 and 10 (excluding one number) and returns the missing number.

Note: The array of numbers will be **unsorted** (not in order). Only one number will be missing.

Input	Output
[1, 2, 3, 4, 6, 7, 8, 9, 10]	5
[7, 2, 3, 6, 5, 9, 1, 4, 8]	10
[10, 5, 1, 2, 4, 6, 8, 3, 9]	7

Problem: 2

Write a C++ program that takes an unsorted array and returns the **nth** smallest integer entered by the user. (the smallest integer is the **first smallest**, the second smallest integer is the **second smallest**, etc).

Note:

1. n will always be ≥ 1 .
2. Each number in the array will be distinct (there will be a clear ordering).
3. Given an out of bounds parameter (e.g. an array is of size k), and you are asked to find the $m > k$ smallest integer, return -1.

Table 5.2

Input	Output
[1, 3, 5, 7] 1	1
[1, 3, 5, 7] 3	5
[7, 3, 5, 1] 2	3

Problem: 3

Mona has created a method to sort an array in ascending order.

Starting from the left of the array, she compares numbers by pairs. If the first pair is ordered [smaller number, larger number], she moves on. If the first pair is ordered [larger number, smaller number], she swaps the two integers before moving on to the next pair. She repeats this process until she reaches the end of the array.

Then, she moves back to the beginning of the array and repeats this process until the entire array is sorted.

If the unsorted array is: [3, 9, 7, 4], she will perform the following steps (note *Swaps* here refers to cumulative swaps):

- She starts with the first pair.
- [3, 9] is in order, move on. Array: [3, 9, 7, 4]. Swaps: 0.
- [9, 7] is not. Swap. Array: [3, 7, 9, 4]. Swaps: 1
- [9, 4] is not. Swap. Array: [3, 7, 4, 9]. Swaps: 2
- Check if the array is sorted. It is not, so start over at the first pair.
- [3, 7] is in order, move on. Array: [3, 7, 4, 9]. Swaps: 2
- [7, 4] is not. Swap. Array: [3, 4, 7, 9]. Swaps: 3.
- [7, 9] is in order, move on. Array: [3, 4, 7, 9]. Swaps: 3.
- Check if the array is sorted. It is. End.

Sorting the array [3, 9, 7, 4] takes her 3 swaps. Write a C++ program that takes an unsorted array from the user and returns the number of swaps it takes for the array to be sorted according to Mona's algorithm as well as the sorted array.

Test Cases

Table 5.10

Input	Output
[5, 4, 3]	3 [3, 4, 5]
[1, 3, 4, 5]	0 [1, 3, 4, 5]
[5, 4, 3, 2]	6 [2,3,4,5]