1. Write a program to input an integer array consisting of n elements and an integer x. Checks if the value x is in the array just entered.
   If x is in the array, print “exist” to the screen. Otherwise, print to the screen “non-exist”.

Std in	Std out

5
1 2 3 4 5
10
	non-exist

5
10 20 30 40 50
30
	exist

2. Write a program, input two strings s_1 and s_2. Check if string s_2 is inside string s_1 or not.

If string s_2 is inside string s_1, print “exist” to the screen. Otherwise, print to the screen “non-exist”.

Note, do not use the string library's support functions. Students must treat the two strings s_1 and s_2 as two arrays. There is a distinction between uppercase and lowercase characters.

Std in	Std out
hello world world	exist
university of science social	non-exist

3. Write a program that inputs an integer array consisting of n "pre-sorted" elements and the value x. Write a program to check if element x is in the array you just entered. **Use binary search.**

If x is in the array, print “exist” to the screen. Otherwise, print to the screen “non-exist”.

Std in	Std out

5
10 20 30 40 50
40
	exist

5
10 20 30 40 50
100
	non-exist

4. Write a program, input an integer array with n elements and the value x. Count the number of occurrences of the value x in the entered array.

Std in	Std out

5
10 50 20 30 30
100
	0

5
10 50 20 30 30
30
	2

5. Write a program to input an integer array of n elements. Print to the screen, the most appearing element in the array. If more than two elements are the result, print all of them.

Std in	Std out
5 10 50 20 30 30	30
7 10 50 20 30 30 50 50	50

6. Bubble sort

1. Write a program to input an array of n integers. Print to the screen the elements in the array have been sorted in order from smallest to largest. Students use Bubble sort.
2. Students indicate the complexity of this algorithm and explain why (briefly), by commenting in the code.

Std in	Std out
5 10 50 20 30 40	10 20 30 40 50

7. Same content as question 6. But with the Quick sort algorithm.

Std in	Std out
5 10 50 20 30 40	10 20 30 40 50

8. Write a program to input two pre-sorted arrays of integers arr_1 and arr_2. Print to the screen, the resulting array is the concatenation of the two entered arrays. Note, the resulting array still has to be sorted from smallest to largest.

Students use the idea of the Merge sort algorithm.

Std in	Std out

5
10 20 30 40 50
2
25 100
	10 20 25 30 40 50 100

2
5 20
2
10 15
	5 10 15 20

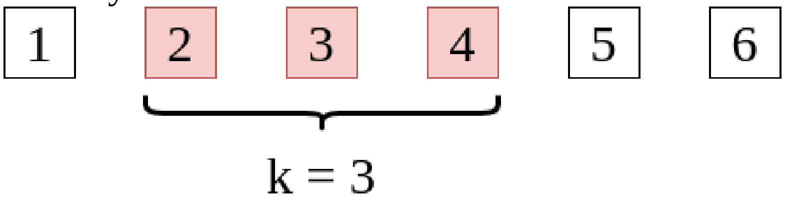
9. Write a program to input an array of n integers. Find the sum of two elements in the array, such that this sum has the maximum value. Print the total value to the screen.

Analyze the complexity of your algorithm and explain the results (briefly) by commenting in the code.

Std in	Std out
5 10 50 20 30 40	90
5 -10 0 -20 -15 0	0

10. Write a program to input an array of n integers. Write a program to find a subarray of k consecutive elements (k < n), such that the sum of this subarray is the largest. If there are multiple subarrays with the same value, print all of them.

Illustrate a sub-array of k consecutive elements:



Std in	Std out

5
10 50 20 30 40
3
	50 20 30

5
-10 0 6 -1 -5
3
	0 6 -1