Question 7.

Write a program to find the second largest element of a given integer array. Can you modify it to find the kth largest element?

In the second-largest() function first the array is sorted using selection sort in descending order.

So that the value stored in the 1st index of the sorted array is the second largest element.

In order to find the kth largest element in the array; after sorting the array in descending order the value stored in k-1th index contains the kth largest element in the array.

Question 8.

Write a program to count the total number of duplicate elements in an unsorted character array in O(n) time complexity

The count_duplicate() function counts the number of character that is repeated in the character array.

First an array char_count is initialized with 256 element and all these

elements are initialized to zero.

Then each element of the character array is iterated through a for loop and each element is compared to each element of the count_char array after its index value is converted into character data type. It is 256 elements signifying all the characters in ASCII.
If the char is similar to the element in the array then array index corresponding to the ASCII value of the character is incremented. such that the count of the elements in the character array is stored in the ASCII value array.

Then in another loop each element of the ASCII value array count_char is checked and if the value of any element is greater than one then the no_of_dupli is incrimented.
so it returs the number of character that is repeated in the character array.

The resultant time complexity is O(n²) since nested for loops are not used.

Question - 9.

Write a program to merge two sorted arrays of the same size to get resultant array which inserted in the reverse order. Analyse the time complexity of your Algorithm.

The time complexity of the Algorithm used is $O(n^2)$ because selection sort is used in this Algorithm to sort the merged array in reverse order.

merging array time complexity : $O(n)$
selection sort complexity $O(n^2)$
reading array complexity $O(n)$
Print array complexity $O(n)$

$$O(n) + O(n^2) + O(n) + O(n)$$

$\Rightarrow$ The resultant time complexity is $O(n^2)$