TASK-1

At first we read from the text file, then separate the size and the array. Here we defined two functions. One is ‘mergesort’ which takes an array and sorts it using the divide and conquer method. The array is recursively split into two parts until the base case is reached.. Then we call the ‘merge’ function to merge the sorted subarray back together as it uses two pointers to traverse the two arrays while comparing elements and merging them in a sorted order. Finally the array is written in the output text file.

TASK-2

At first we read from the text file, then separate the size and the array. The ‘mergesort’ function takes an array as an input, it recursively splits the array into two parts until the base case is reached and compares the maximum value. Then it returns the maximum value. Finally the maximum value is written in the output text file.

TASK-3

At first we read from the text file, then separate the total and the array. The ‘mergesort’ function iterates through the list to count how many aliens are out of order following the mentioned condition. If the condition matches then return 1. Finally the total count is written in the output text file.

TASK-4

At first we read from the text file. First dividing the array recursively in left and right. Then we find the maximum element with the condition possible for both sub arrays. Also, we checked if any maximum element is possible from the total element from both arrays. This is done by adding the maximum element of the left array and maximum squared value of the right array and finally returning the maximum value among those three. Finally we write the result in the output text file.

TASK-5

At first we read from the text file, then separate the size and the array. The ‘quicksort’ function takes an array, first index and last index as parameters. It recursively calls the partition function and sorts the sub array. Partition function chooses a pivot and reorganizes the elements in the array such that all elements less than or equal to the pivot are on the left side and all elements greater than the pivot are on the right side and it returns the index of the pivot. Then the sorted array is written on the output text file.

TASK-6

At first we read from the text file, then separate the total elements and the list and append all the query in another list. Then iterates through the query list and calls the ‘find’ function.

It takes an array and a value from a query list as an input. It finds the kth smallest element of the array by recursively calling the partition function, comparing the pivot index and returns the element. Finally we write the result in the output text file.