**Q. Given an array of n positive integers. Find the k-th largest element in the array and then find the maximum sum of a subsequence such that no two numbers in the subsequence are adjacent in the array, and the subsequence must include the k-th largest element.**

***For example: arr[] = {5, 5, 10, 100, 10, 5} and k = 2. Output is 25.***

***Here, the 2nd largest element is 10. The subsequence {5, 10,10} includes 10 and gives the maximum sum of 25, with no two elements being adjacent.***

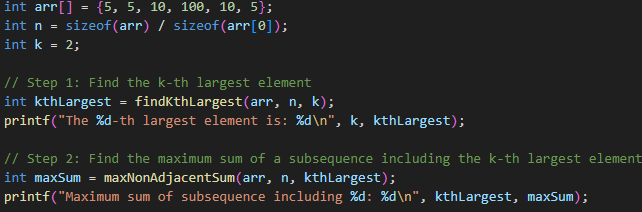
***Another example for more clarification: arr[] = {4, 1, 3, 9, 2, 8} and k = 2. Output is 21.***

***The 2nd largest element is 8. The subsequence {4,9,8) includes 8 and max sum 21 and no two elements are adjacent.***

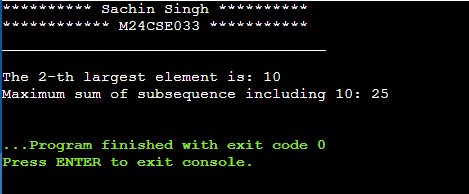
**Explanation:**

1. **findKthLargest**: It will return the k-th largest element after sorting the array in decreasing order.
2. **maxNonAdjacentSum**: This will return the maximum sum of a subsequence where no two elements are adjacent, and it will certainly include the k-th largest element.
3. **Main Function**: We have declared an array, arr[], and have found the k-th largest element using the function, findKthLargest. Then we recursively call the method maxNonAdjacentSum, which looks to find the maximum sum of a subsequence including the k-th largest element. Finally, print the result.

**Example:**

****

**Output:**

****