Experiment No. 5

Aim: Demonstrate the use of one-dimensional arrays to solve a given problem.

1 A : The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from the unsorted part and putting it at the beginning. The algorithm maintains two sub-arrays in a given array.

1) The sub-array which is already sorted.
2) Remaining sub-array which is unsorted.

Every iteration of selection sort, the minimum element (considering ascending order) from the unsorted sub-array is picked and moved to the sorted sub-array.

Following example explains the above steps:

arr[] = 64 25 12 22 11
// Find the minimum element in arr[0...4]
// and place it at beginning
11 25 12 22 64
// Find the minimum element in arr[1...4]
// and place it at beginning of arr[1...4]
11 12 25 22 64
// Find the minimum element in arr[2...4]
// and place it at beginning of arr[2...4]
11 12 22 25 64
// Find the minimum element in arr[3...4]
// and place it at beginning of arr[3...4]
11 12 22 25 64

B. Perform search of a particular element on the above array using binary search.
Binary Search will search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty. We basically ignore half of the elements just after one comparison.

1. Compare x with the middle element.
2. If x matches with middle element, we return the mid index.
3. Else If x is greater than the mid element, then x can only lie in right half subarray after the mid element. So we recur for right half.
4. Else (x is smaller) recur for the left half.

Batch 1:Delete duplicates from array.
sample i/p : [1 2 1 4 3 2 1 3 2 1 4 5 7 6 9 9]
o/p: [1 2 4 3 5 7 6 9]

Batch 2: Circular Array Rotation means rotating the elements in the array where one rotation operation moves the first element of the array to the last position and shifts all remaining elements to the left.

Here, we are given an unsorted array and our task is to perform the circular rotation by n number of rotations where n is a natural number.
Initial Array: [1 2 3 4 5]

After one rotation : [2 3 4 5 1]

After two rotation : [3 4 5 1 2]

Batch 3: Write a program in C to calculate frequency count of each number in ARRAY.
Suppose input array is [1 2 1 4 5 2 3 1 4 5 6 2 2 3]
output : 1 is appearing 3 times , 2 is appearing 4 times , 4 is appearing 2 times….5 is appearing 2 times , 3 is appearing 2 times , 6 is appearing 1 times

Batch 4: insert and delete element from array.
insert demo: sample i/p: [2 4 5 7 8 4 3 6 7]
insert 3 at 4th position

o/p: [2 4 5 3 7 8 4 3 6 7]
delete demo : sample i/p: [2 4 5 7 8 4 3 6 7]
delete 7th element
o/p: [2 4 5 7 8 4 6 7] 3 gets deleted as it is 7th element