

LINEAR & BINARY SEARCHAIM

write a menu driven c program to implement linear search and Binary search

ALGORITHM

1. start
2. initialise int arr[20] and variable r
3. create read function to read the elements of the array with int x as parameter
4. create display function to display the elements
5. create linear function with x as parameter (element to be searched)
 - 5.1 set flag = 0
 - 5.2 loop until arr, check if $arr[i] = x$
 - 5.3 if yes set flag = 1 and break.
 - 5.4 if flag = 0, print element not found
6. create function binary with element to be searched as parameter
 - 6.1 set beg = 0, end = 0
 - 6.2 loop until $i \leq n-1$ ($n \rightarrow$ total elements)
 - 6.2.1 loop until $j \leq n-i-1$
 - 6.2.2. using a variable temp, sort array and print the sorted array

6.3 set $end = n - 1$

6.4 while $beg \leq end$

6.4.1 $mid = (beg + end) / 2$

6.4.2 if $(arr[mid] == x)$, print element - bound

6.4.2.1 $beg = 1$

6.4.2.2 break

6.4.3 else if $arr[mid] > x$

set $end = mid - 1$

6.4.4 else set $beg = mid + 1$

6.5 if $beg = 0$, print element not bound

7 in main,

7.1 print menu

7.2 read choice, and call respective functions

8 stop

CONCLUSION

The program was executed correctly and output has been verified

BOBBLE, INSERTION & SELECTION SORTAIM

write a menu-driven program to implement Bubble sort, insertion sort and selection sort.

ALGORITHM

1. start
2. globally declare integers i, j, n , temp and array $a[10]$;
3. declare void function menu()
 - 3.1 read n , total number of elements
 - 3.2 set $i = 0$, loop until $i < n$.
 - 3.2.1 read $a[i]$
4. declare function display() to display the array
5. declare void function bubbleSort()
 - 5.1 set $i = 0$, loop until $i < n$
 - 5.1.1 set $j = 0$, loop $j < n - i - 1$
 - 5.1.1.1 if $(a[j] > a[j+1])$
→ swap $a[j]$ and $a[j+1]$
 - 5.2 call display function
6. declare void function insertionSort()
 - 6.1 loop until $i < n$
 - 6.1.1 set temp = $a[i]$, $j = i - 1$

6.1.2 while $(temp < a[j])$ & $(j > 0)$

6.1.2.1 $a[j+1] = a[j]$

6.1.2.2 $j--$

6.1.3 $a[j+1] = temp$

6.2 call display function

7 declare integer function smallest with parameter k

7.1 set $pos = k$, $small = a[k]$, $i = k+1$

7.2 loop until $i < n$

7.2.1 if $(a[i] < small)$

set $small = a[i]$, $pos = i$

7.3 return pos

8 declare void function selection sort (c)

8.1 set $k = 0$, loop until $k < n$

8.1.1 $pos = \text{smallest}(k)$

8.1.2 $temp = a[k]$

8.1.3 $a[k] = a[pos]$

8.1.4 $a[pos] = temp$

8.2 call display function.

9 in main,

9.1 print menu

9.2 read choice

9.3 using switch case call the resp function

stop

CONCLUSION

The program has been executed correctly and output has been verified.