COMPUTER PROGRAMING

EX NO - 4

Name: Kakavakam Jaswanth Sai

Roll no: CH.EN.U4CSE20130

Section: CSE-B

Subject code: 19CSE102 LAB

EX NO -4: PROGRAMS USING RECURSIVE FUNCTION

1. LINEAR SEARCH USING RECURSIONS.

Aim:

To Write a program in C to find an element using linear search by recursive Functions.

Algorithm:

MAIN:

Step 1: Start

Step 2: Declare an array Arr[100] of integer datatype,n,sele.

Step 3: input the value of n .

Step 4: for (int i=0; i<n; i++)

Get the input of Arr[i]

Step 5: input the value of “sele” the element to search.

Step 6: Call the function linear (Arr ,0, n-1, sele) and store the value to c.

Step 7: if (c! =-1) print that element found in the location of index c.

Step 8: else print element not found.

Step 9: Stop.

Linear (int Array [], int findex, int lindex, int search)

Step 1: Start

Step 2: if (lindex < findex)

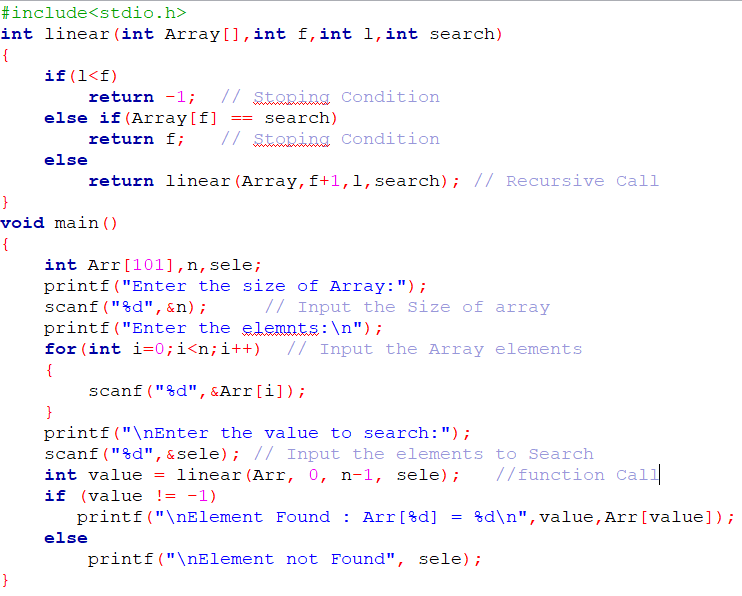
Return ( -1);

Step 3: else if (Array[findex] == search)

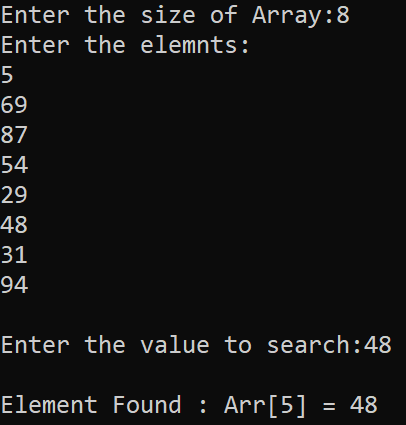
Return findex;

Step 4: else return linear (Array, findex+1, lindex, search)

Program:



Output:



Result:

Thus, the program to find an element using linear search by recursive Functions in C language

has been executed and verified successfully.

1. BINARY SEARCH USING RECURSIONS.

Aim:

To Write a program in C to find an element using binary search by recursive Functions.

Algorithm:

MAIN:

Step 1: Start

Step 2: Declare an array Arr[100] of integer datatype,n,sele.

Step 3: input the value of n .

Step 4: for (int i=0; i<n; i++) // elements should be sorted

Get the input of Arr[i]

Step 5: input the value of “sele” the element to search.

Step 6: Call the function bisearch (Arr ,0, n-1, sele) and store the value to c.

Step 7: if (c! =-1) print that element found in the location of index c.

Step 8: else print element not found.

Step 9: Stop.

bisearch (int Array [], int low, int high, int search)

Step 1: Start

Step 2: if (high >= low)

{

1. Calculate Average of low and high and store it to mid.
2. If (Array[mid] == search) Return mid;
3. If (Array[mid] > search)

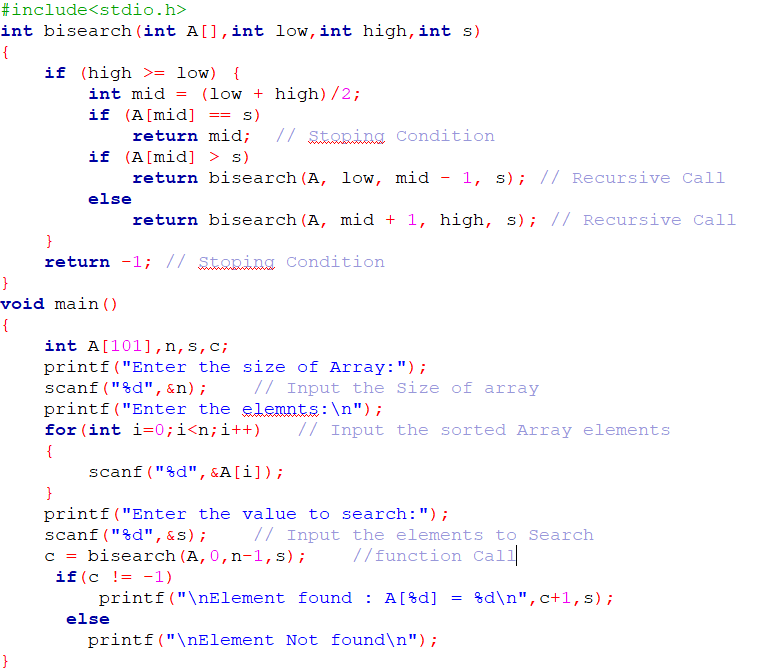
return bisearch (A, low, mid - 1, s);

else

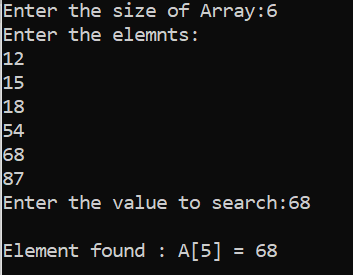
return bisearch (A, mid + 1, high, s);

Step 3: Return ( -1);

Program:



Output:



Result:

Thus, the program to find an element using binary search by recursive Functions in C language

has been executed and verified successfully.

1. SELECTION SORT USING RECURSIONS.

Aim:

To Write a program in C to Sort An Array using Selection sort by recursive Functions.

Algorithm:

MAIN:

Step 1: Start

Step 2: Declare an array Arr [100] of integer datatype,n.

Step 3: input the value of n .

Step 4: for (int i=0; i<n; i++) // elements should be sorted

Get the input of Arr[i]

Step 5: Call the function Selection (Arr, 0, 0, n, 1);

Step 6: Print the Array “Arr”.

Step 9: Stop.

Selection (int list[], int i, int j, int size, int flag)

Step 1: Start

Step 2: Declare an integer temp.

Step 3: if (i < size - 1)

{ if (flag) {

j = i + 1;

}

if (j < size) {if (list[i] > list[j]) { temp = list[i];

list[i] = list[j];

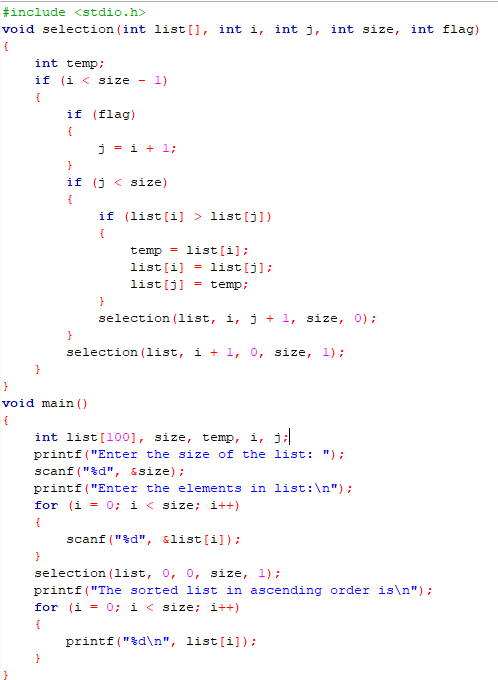
list[j] = temp;

}

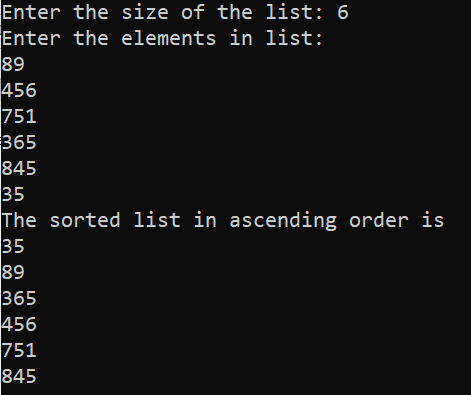
selection(list, i, j + 1, size, 0);}

selection(list, i + 1, 0, size, 1);}

Program:



Output:



Result:

Thus, the program to Sort An Array using Selection sort by recursive Functions in C language

has been executed and verified successfully.