PRACTICAL:-01

AIM:- WAP and algorithm of linear search and binary search.

//Linear search algorithm

Linear_Search (DATA, n , ITEM , LOC)

1. Set k=1 and LOC=0

2. Repeat steps 3 and 4 while LOC=0 and k<=n

3. If ITEM = Data[k], then set LOC=k

4. Set k = k+1

[End of loop]

5. If LOC=0 then

Write: ITEM is not in the array

Else

Write: LOC is the location of ITEM

[End of if structure]

6. Exi

// COMPLIXTY

Best Case O(1)

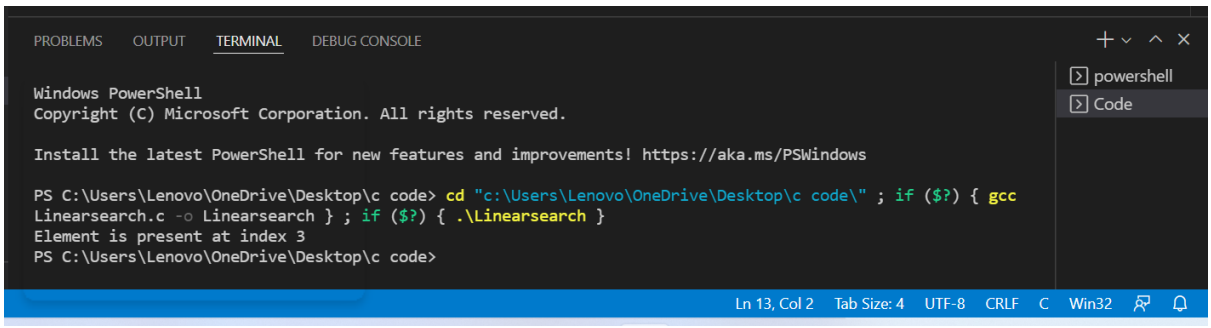
Average Case O(n)

Worst case O(n)

//PROGRAM:-

```
#include <stdio.h>

int search(int arr[], int n, int x)
```

```
{
        int i;
        for (i = 0; i < n; i++)
                if (arr[i] == x)
                        return i;
        return -1;
}
int main(void)
{
        int arr[] = { 2, 3, 4, 10, 40 };
        int x = 10;
        int n = sizeof(arr) / sizeof(arr[0]);
int result = search(arr, n, x);
        (result == -1)
                ? printf("Element is not present in array")
                : printf("Element is present at index %d", result);
        return 0;
}
//OUTPUT
```



//BINARY SEARCH ALGORITHM

1. Set BEG:= LB, END:=UB & MID= INT((BEG+END)/2)

2. Repeat steps 3 & 4 while ((BEG <= END) and (DATA[MID] ≠ ITEM)) do

3. if (ITEM < DATA[MID]) then

Set END = MID – 1

else

Set BEG = MID + 1

[Endif]

4. Set MID = INT((BEG + END) / 2)

(End of Step 2 loop)

5. If DATA[MID]=ITEM

then Set LOC = MID

Else

Set LOC= NULL endif

6. Exit

// COMPLEXITY

Best Case O(1)

Average Case O(logn)

Worst Case O(logn

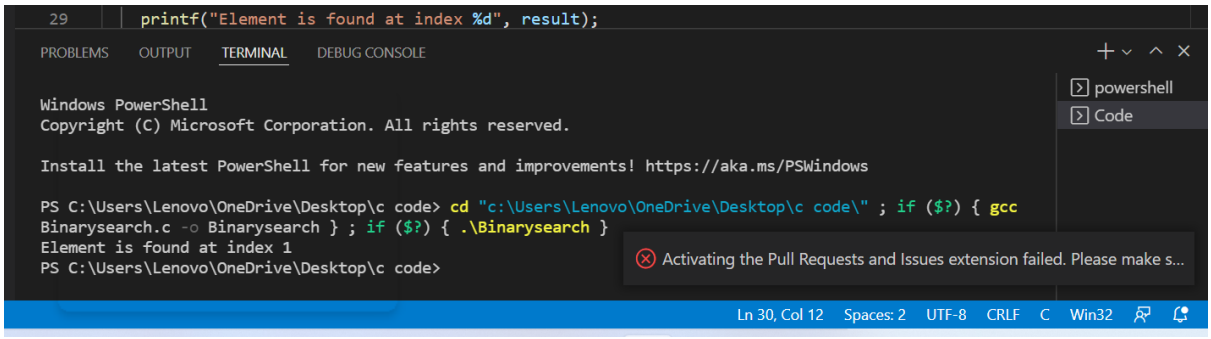
//PROGRAM

```
#include <stdio.h>

int binarySearch(int array[], int x, int low, int high) {

  while (low <= high) {
```

```
    int mid = low + (high - low) / 2;

    if (array[mid] == x)
      return mid;

    if (array[mid] < x)
      low = mid + 1;

    else
      high = mid - 1;
  }

  return -1;
}

int main(void) {
  int array[] = {3, 4, 5, 6, 7, 8, 9};
  int n = sizeof(array) / sizeof(array[0]);
  int x = 4;
  int result = binarySearch(array, x, 0, n - 1);
  if (result == -1)
    printf("Not found");
  else
    printf("Element is found at index %d", result);
  return 0;
```

}

//OUTPUT


29 printf("Element is found at index %d", result);
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Lenovo\OneDrive\Desktop\c code> cd "c:\Users\Lenovo\OneDrive\Desktop\c code\" ; if ($?) { gcc Binarysearch.c -o Binarysearch } ; if ($?) { .\Binarysearch }
Element is found at index 1
PS C:\Users\Lenovo\OneDrive\Desktop\c code>
Activating the Pull Requests and Issues extension failed. Please make s...
Ln 30, Col 12 Spaces: 2 UTF-8 CRLF C Win32