**Name:** Purvi Singhvi

**Enroll. No.:** 180099

**PROGRAM – 8**

**Aim:** Write an algorithm and program to implement interpolation search.

**Algorithm:**

**1.** L=0, h=n-1

**2.** while (A[h] != A[L] && (x>=A[L]) && (x<=A[h]) )

**3.** pos = L + ( ((x-A[L])/(A[h]-A[L])) \* (h-L) )

**4.** if(x<A[pos])then

**5.** h = pos-1

**6.** else if (x>A[pos])then

**7.**  L= pos+1

**8.** else

**9.** Return(pos)

**10.** STOP

**Source Code:**

#include<stdio.h>

#include<conio.h>

int interpolationSearch(int arr[], int n, int x)

{

int lo = 0, hi = (n - 1);

while (lo <= hi && x >= arr[lo] && x <= arr[hi])

{

if (lo == hi)

{

if (arr[lo] == x)

return lo;

return -1;

}

int pos = lo + (((double)(hi-lo) / (arr[hi]-arr[lo]))\*(x - arr[lo]));

if (arr[pos] == x)

return pos;

if (arr[pos] < x)

lo = pos + 1;

else

hi = pos - 1;

}

return -1;

}

void main()

{

int arr[15],n,x,i;

clrscr();

printf("Enter size of array:");

scanf("%d", &n);

printf("Enter elements of array :");

for(i=0;i<n;i++)

scanf("%d", &arr[i]);

printf("Enter the element to search: ");

scanf("%d", &x);

int index = interpolationSearch(arr, n, x);

if (index != -1)

printf("Element found at index %d", index);

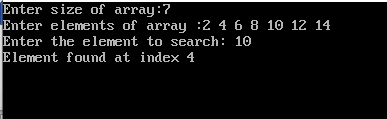
else

printf("Element not found.");

getch();

}

**Output:**



**Complexity:**

* Worst case time complexity: O(N)
* Average case time complexity: O(log log N)
* Best case time complexity: O(1)