**Name:** Priyal Jain

**Enrollment No.:** 180091

**PROGRAM-8**

**Aim:** Write an algorithm and program to search a number using Interpolation Search technique.

**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 interpoln\_Search(int a[], int s, int m)

{

int l = 0, h = (s - 1);

while (l<= h && m >= a[l] && m <= a[h])

{

if (l == h)

{

if (a[l] == m)

return l;

return -1;

}

int positn = l + (((double)(h-l) / (a[h]-a[l]))\*(m - a[l]));

if (a[positn] ==m)

return positn;

if (a[positn] < m)

l = positn + 1;

else

h = positn - 1;

}

return -1;

}

void main()

{

int a[15],s,m,i;

clrscr();

printf("Enter the size of an array:");

scanf("%d", &s);

printf("Enter the elements of an array :");

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

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

printf("Enter the element that has to be searched: ");

scanf("%d", &m);

int index = interpoln\_Search(a, s, m);

if (index != -1)

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

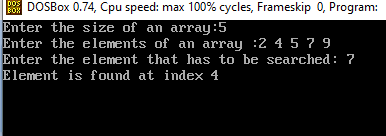
else

printf("Element is not found.");

getch();

}

**Output:**



**Complexity:**

Best Case: O(log log n)

Worst case: O(n)