**Program – 8**

**AIM – Write an algorithm and program to implement interpolation search.**

**Algorithm:**

Step 1 − Start searching data from middle of the list.

Step 2 − If it is a match, return the index of the item, and exit.

Step 3 − If it is not a match, probe position.

Step 4 − Divide the list using probing formula and find the new midle.

Step 5 − If data is greater than middle, search in higher sub-list.

Step 6 − If data is smaller than middle, search in lower sub-list.

Step 7 − Repeat until match.

**Source Code:-**

#include<bits/stdc++.h>

using namespace std;

intinterpolationSearch(intarr[], 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;

}

intpos = lo + (((double)(hi - lo) /

(arr[hi] - arr[lo])) \* (x - arr[lo]));

if (arr[pos] == x)

returnpos;

if (arr[pos] < x)

lo = pos + 1;

else

hi = pos - 1;

}

return -1;

}

int main()

{

intarr[] = {10, 12, 13, 16, 18, 19, 20, 21,

22, 23, 24, 33, 35, 42, 47};

intn = sizeof(arr)/sizeof(arr[0]);

int x = 18; // Element to be searched

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

if (index != -1)

cout<< "Element found at index " << index;

else

cout<< "Element not found.";

return 0;

}