**PROGRAM-8**

**AIM-** Write an algorithm and program to implement interpolation search

ALGORITHM-

mid = Lo + ((Hi - Lo) / (A[Hi] - A[Lo])) \* (X - A[Lo])

where −

A = list

Lo = Lowest index of the list

Hi = Highest index of the list

A[n] = Value stored at index n in the list

**Source Code-**

#include<stdio.h>

#define MAX 10

// array of items on which linear search will be conducted.

int list[MAX] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44 };

int find(int data) {

int lo = 0;

int hi = MAX - 1;

int mid = -1;

int comparisons = 1;

int index = -1;

while(lo <= hi) {

printf("\nComparison %d \n" , comparisons ) ;

printf("lo : %d, list[%d] = %d\n", lo, lo, list[lo]);

printf("hi : %d, list[%d] = %d\n", hi, hi, list[hi]);

comparisons++;

// probe the mid point

mid = lo + (((double)(hi - lo) / (list[hi] - list[lo])) \* (data - list[lo]));

printf("mid = %d\n",mid);

// data found

if(list[mid] == data) {

index = mid;

break;

} else {

if(list[mid] < data) {

// if data is larger, data is in upper half

lo = mid + 1;

} else {

// if data is smaller, data is in lower half

hi = mid - 1;

}

}

}

printf("\nTotal comparisons made: %d", --comparisons);

return index;

}

int main() {

//find location of 33

int location = find(33);

// if element was found

if(location != -1)

printf("\nElement found at location: %d" ,(location+1));

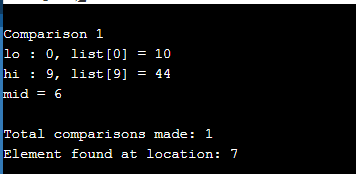
else

printf("Element not found.");

return 0;

}

**OUTPUT-**

****