// C++ program to implement Jump Search

#include<iostream>

#include<cmath>

using namespace std;

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

{

// Finding block size to be jumped

int step = sqrt(n);

// Finding the block where element is

// present (if it is present)

int prev = 0;

while (arr[min(step, n)-1] < x)

{

prev = step;

step += sqrt(n);

if (prev >= n)

return -1;

}

// Doing a linear search for x in block

// beginning with prev.

while (arr[prev] < x)

{

prev++;

// If we reached next block or end of

// array, element is not present.

if (prev == min(step, n))

return -1;

}

// If element is found

if (arr[prev] == x)

return prev;

return -1;

}

// Driver program to test function

int main()

{

int arr[] = { 0, 1, 1, 2, 3, 5, 8, 13, 21,

34, 55, 89, 144, 233, 377, 610 };

int x = 55;

int n = sizeof(arr) / sizeof(arr[0]);

// Find the index of 'x' using Jump Search

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

// Print the index where 'x' is located

cout << "\nNumber " << x << " is at index " << index;

return 0;

}

// Contributed by nuclide

