Jump Search

```
import math

def jumpSearch( arr , x , n ):

    # Finding block size to be jumped
    step = math.sqrt(n)

    # Finding the block where element is
    # present (if it is present)
    prev = 0
    while arr[int(min(step, n)-1)] < x:
        prev = step
        step += math.sqrt(n)
        if prev >= n:
            return -1

    # Doing a linear search for x in
    # block beginning with prev.
    while arr[int(prev)] < x:
        prev += 1

        # 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[int(prev)] == x:
        return prev

    return -1

# Driver code to test function
arr = [ 0, 1, 1, 2, 3, 5, 8, 13, 21,
    34, 55, 89, 144, 233, 377, 610 ]
x = 55
n = len(arr)
```

```
# Find the index of 'x' using Jump Search
index = jumpSearch(arr, x, n)

# Print the index where 'x' is located
print("Number" , x, "is at index" ,"%.0f"%index)
```