Problem :

Given an array of some length and an element to search in a given array using Binary Search recursive algorithm**.**

if we use a simple linear search algorithm, the time complexity for this algorithm will be **o(n)** but if we use the Binary Search algorithm the time complexity for this algorithm will be reduced to **O(log(n)).**

Assumption: Array is already sorted

Input:

- The first line of the input takes two values **n** and **x,** where n is the length of the array and x is the element to find.
- The second line of input takes **an n-size array.**

Output:

- the program returns an **index** at which element is present else return **-1**

Pseudocode:

- Divide the array into two parts and take the middle element.
- Compare x with the middle element.
- if x matches with the middle element then the middle is the element, we need.
- else if x is greater than the middle element, then x lies only on the right side of the middle element.
- else x lies on the left side of the middle element.
- Now, search in only the part of the array left call that part recursive.

language Used: python3

Filename: BinarySearch.py

Command to run a python program in Linux machine: python3 BinarySearch.py

code:

```
def binarySearch(array, start, end, x):
    """
```

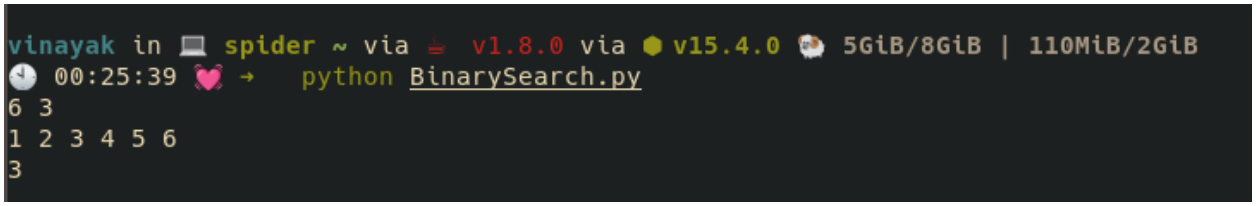
```
    binarySearch function:
        this function is used to search an element in
        a given array
    """
    if end >= start:
        # finding middle index
        mid = start + (end - start) // 2

        if array[mid] == x:
            # x matches with the middle element then return mid
            return mid
        elif array[mid] > x:
            # if x is less shift last index to middle
            return binarySearch(array, start, mid - 1, x)
        else:
            # if x is greater shift fisrt index to middle
            return binarySearch(array, mid + 1, end, x)
    else:
        # if element not found return -1
        return -1


def main():
    """
    main function:
        this function is used for calling and running
        binary search program
    """
    n, x = map(int, input().split())
    array = list(map(int, input().split()))
    index = binarySearch(array, 0, n - 1, x)
    print(index+1)


if __name__ == "__main__":
    main()
```

Output:



References:

- Modified Binary Search Algorithm by Ankit R. Chadha, Tanaya Mokashi, and Rishikesh Misal.
- Introductions to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.