SEARCH IN ROTATED SORTED ARRAY

☆ In this problem we are given an array which was initially sorted array but then rotated. Our job is to find position of target element.

Brute force solution is to use Linear Search but we need to use Binary Search. Because we rotated the array, only one half remains sorted now while the other one isnt. At each iteration we find the half that is sorted look for our element there. If it is there, we continue with that half else we continue with the other half.

Pseudocode :

```
searchInRotatedSortedI(arr, N, a) {
    low = 0
    high = N - 1
    while(low <= high) {
        mid = (low + high)/2
        if (arr[mid] == a) {
            return mid
        } else if(arr[mid] < arr[high]) {
            if (a <= arr[high] && a >= arr[mid])
```

```
                low = mid
            } else {
                high = mid
            }
        } else {
            if(a <= arr[mid] && a >= arr[low]){
                high = mid
            } else {
                low = mid
            }
        }
    }
    return -1
}
```