SHUBHAM GUPTA

180749

Theo Ass 1

6) pseudocode:

```
q6 (A, B) {
    l ← 0
    r ← m - 1
    flag ← 0
    while ( r > l+1 ) {
        mid ← ⌊(r+l)/2⌋
        if ( A[mid] == B[m - mid - 1] ) {
            flag ← 1
            return mid
        } else if ( A[mid] > B[m - mid - 1] ) {
            r ← mid - 1
        } else if ( A[mid] < B[m - mid - 1] ) {
            l ← mid
        }
    }
    if ( flag == 0 ) {
        return -1
    }
}
```

algorithm :– l is left pointer (0) of A & r is right pointer of B (m-1), now we compare mid value of A & B, we have 3 cases:

C I : A[mid] = B[mid], we get what we need, flag = 1 and exit the code

C II : A[mid] > B[mid], set l to (mid), and run the loop again

C III : A[mid] < B[mid], set r to (mid - 1), and run the loop again time Complexity : $O(\log m)$

proof of correctness: let's suppose we haven't find the sol^n & we are running i^{th} iteration.

after $(i-1)^{th}$ iteration common number will be to the right of previous l in A and to the left of previous r in B

now we have 3 cases:

C I : $A[mid] == A[m-1-mid]$
$mid = k$ solution is found.

C II: $A[mid] > B[m-mid-1]$
here we do not need to $A[mid] \ldots A[n]$ since the values of $A[mid]$ ↑es with ↑ in mid & $B[m-mid-1]$ ↓, so ↓ value of mid we discard right half and get $A[mid_{new}] < A[mid]$ and $B[m-1-mid_{new}] > B[m-mid-1]$

C III: $A[mid] < B[m-mid-1]$
here we do not need anything to the left of current mid. thus $B[m-mid-1]$ will be greater we discard the left half of A and make our new mid on the right side of A,

Here, we can say that with these optimal conditions correctness can be proved.