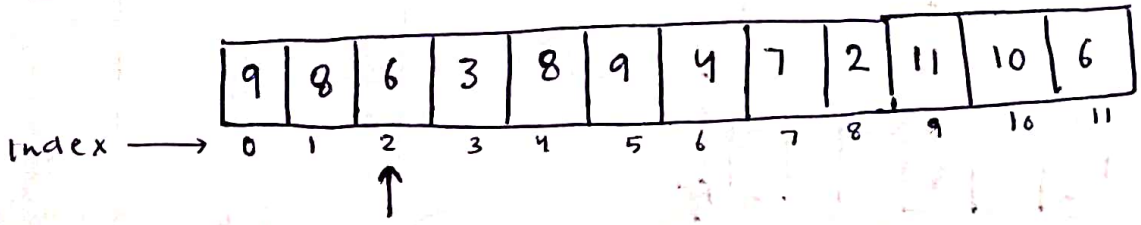
First Index OF Occurance



6 COMES FIRST AT INDEX 2

∴ Return 2

∴ d = data

Expectation

fi (arr, 0, d)

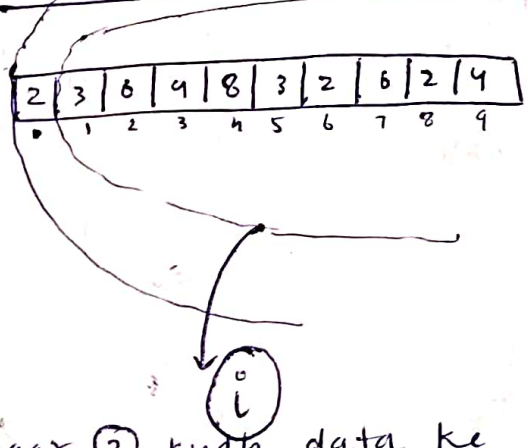
∴ Yeh 0 se end tak purre array ko check krega! Aur (data) ki pheli occurance milegi vo hume return krega!

Faith

fi (arr, 1, d)

∴ Yeh 1 se end tak ki (d) ki pheli occurance return krega!

Expectation meet Faith



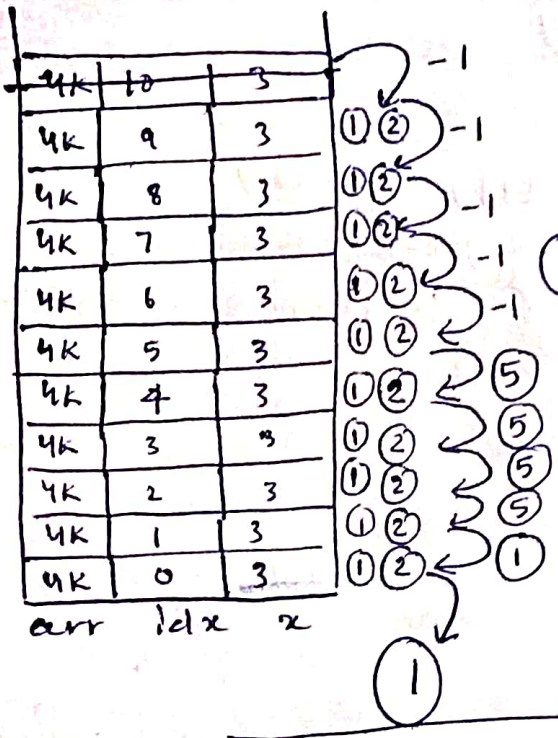
Agar (2) kuch data ke barabar hua toh (i) change hoga! aur (2) ka index (i) me save hoga!

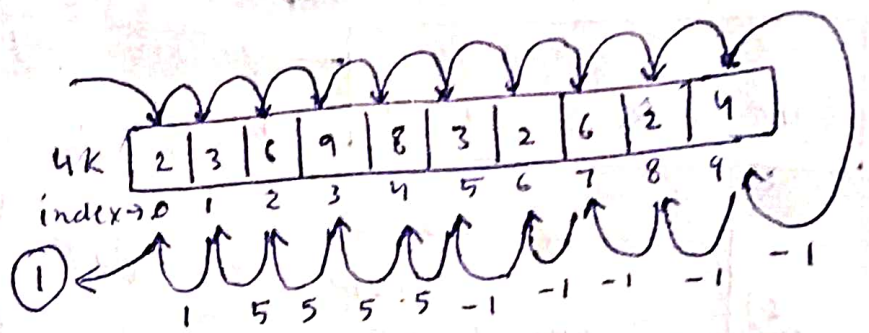
```
psvm(S[]a){
Scanner s= new Scanner(System.in);
 int n = s.nextInt();
int [] arr = new int[n];

for(int i=0; i<arr.length; i++){
    arr[i] = s.nextInt();
}
int d= s.nextInt();
int fi = firstIndex(arr, 0, d);
System.out.println(fi);
}
```

```
ps int firstIndex(int[]arr, int idx,
//BASE CASE                  int x){
int fiisa = firstIndex(arr, idx+1, x);   (1)
if(arr[idx]==x){                          (2)
   return idx;
}else{
   return fiisa;
}
}
```

Base Case ←

```
if(idx == arr.length)
    return -1;
```





Iss CODE me humne comparison baad me kiya toh hume all the way last tak jana pada!

Agar hum comparison phele krlety toh phele hi detect krlety!

BASE CASE

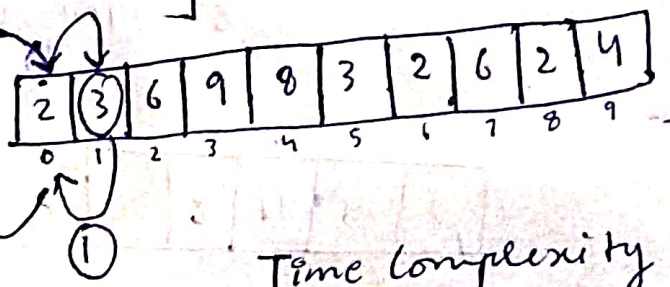
```
if (idx == arr.length)
return -1;
```

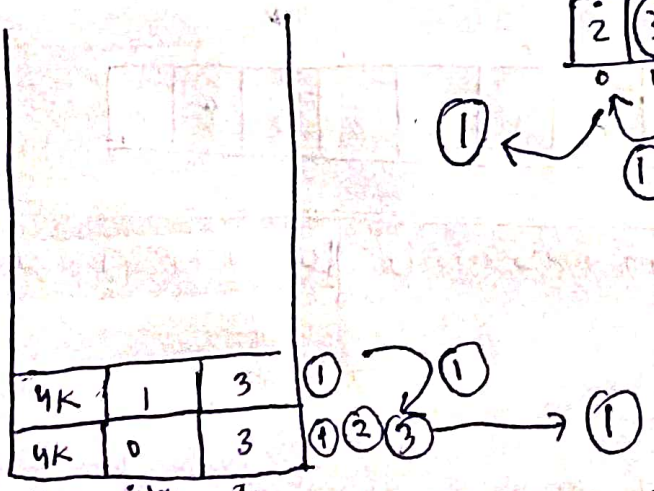
Efficient Way of Code

```
int firstIndex (int [] arr, int idx, int x)
{ if (idx == arr.length) {
      return -1;
  }
  if (arr [idx] == x) {
      return idx;
  } else {
  int fiisa = firstIndex (arr, idx+1, x);
  return fiisa ;
  }
}
```

(1) (2) (3)

∴ Phele comparison krengy Agar barabar nikla toh (idx) return krdengy Agar barabar nahi nikala toh agge dudengy!





Time Complexity : O(n)

Space Complexity : O(1)

Stack Space → O(n)