

● All Indices of An Array

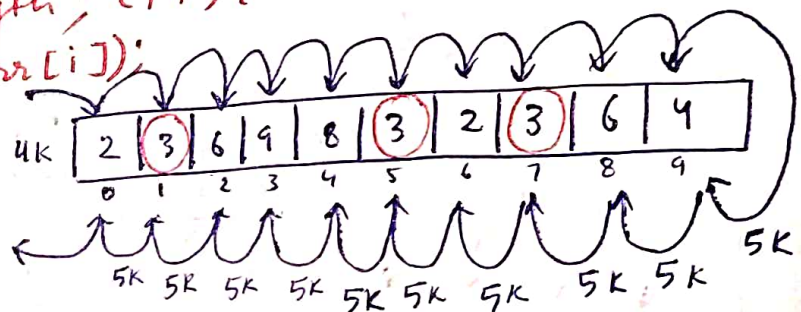
```

p s v m (s c j a) {
    BufferedReader br = new BufferedReader(new InputStreamReader
                                           (System.in));
    int n = Integer.parseInt(br.readLine());
    int[] arr = new int[n];

    for (int i = 0; i < n; i++) {
        arr[i] = Integer.parseInt(br.readLine());
    }
    int x = Integer.parseInt(br.readLine());
    int[] iarr = allIndices(arr, x, 0, 0);

    if (iarr.length == 0) {
        System.out.println();
        return;
    }
    for (int i = 0; i < iarr.length; i++) {
        System.out.println(iarr[i]);
    }
}

```



BASE CASE FOUND			
X	4K	3	10
	4K	3	9
	4K	3	8
	4K	3	7
	4K	3	6
	4K	3	5
	4K	3	4
	4K	3	3
	4K	3	2
	4K	3	1
	4K	3	0
	4K	3	0

arr x idx fsf
↓
found so far

Diagram illustrating the search process for the value 3 in the array. The search starts at index 0 and moves forward. When the value 3 is found at index 1, the search continues. When the value 3 is found again at index 5, the search continues. When the value 3 is found again at index 7, the search continues. The search ends when the value 3 is found at index 10. The search process is shown with arrows and labels indicating the current index and the value being searched for.

Jab value (x) ke equal ho, toh index bhi badaiyen aur fsf bhi badaiyen!

Jab value (x) ke equal nahi ho kaval, idx badaiyen!

Jitne fsf milenge utne size ka array banega! aur base case vo new array return hogi!

Wapis aty hui 5K (new) array ko fill krengy check krty krtty!


```

P.s int[] allIndices (int[] arr, int x, int idx, int fsf)
{
    if (idx == arr.length) {
        return new int[fsf];
    }

```

```

    if (arr[idx] == x) { (1) → call
        int[] iarr = allIndices (arr, x, idx+1, fsf+1); (1.1)
        iarr[fsf] = idx; (1.2)
        return iarr; (1.3)
    }

```

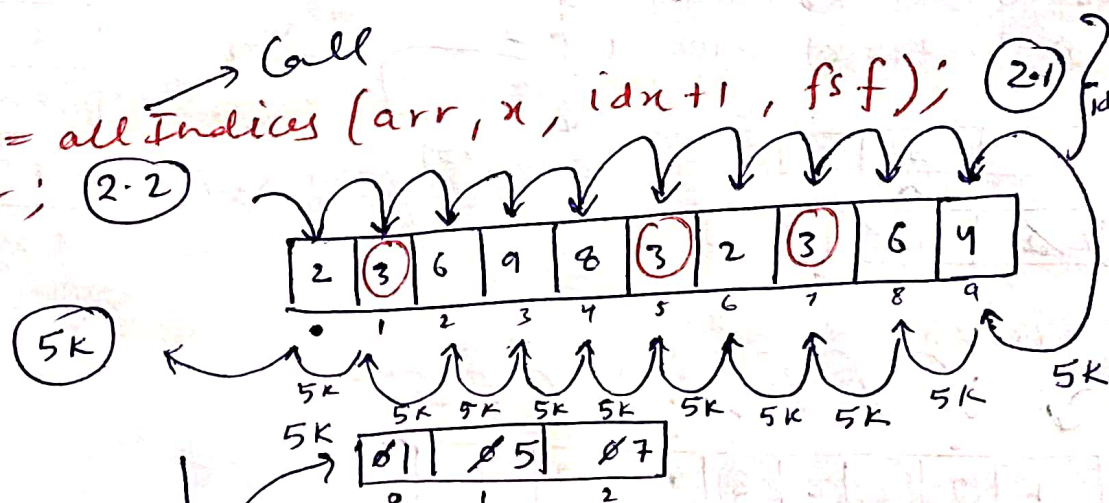
} idx → ↑
fsf → ↑

```

    else {
        int[] iarr = allIndices (arr, x, idx+1, fsf); (2.1)
        return iarr; (2.2)
    }
}

```

} idx → ↑



BASE CASE FOUND			
arr	x	idx	fsf
4K	3	10	3
4K	3	9	3
4K	3	8	3
4K	3	7	2
4K	3	6	2
4K	3	5	1
4K	3	4	1
4K	3	3	1
4K	3	2	1
4K	3	1	0
4K	3	0	0

Diagram illustrating the recursive process. The array is [2, 3, 6, 9, 8, 3, 2, 3, 6, 4]. Elements are mapped to boxes labeled '5K' with values 1, 5, 7. Arrows indicate the flow of data and the recursive calls.

fsf → 2 pe 7 daal do
iarr[2] = 7
1 pe 5 daal do
iarr[1] = 5
0 pe 1 daal do
iarr[0] = 1

5K [1 | 5 | 7]