

Input: [3,2,4], target = 6

Output: [1,2]

Code: [LP_Code3.java](#)

Output

```
Enter the number of elements you want to store: 5
Enter the elements of the array:
2 7 4 1 6
Enter the target element of the array:
9
The output array is: [0,1]
```

```
Enter the number of elements you want to store: 3
Enter the elements of the array:
3 2 4
Enter the target element of the array:
6
The output array is: [1,2]
```

Approach:

We would store every element in the hashmap in the form of (array[index], index). The idea is we will traverse the original array and will check in map if the remaining element i.e. target - currentElement is present in hashmap then we found our answer.

The answer would be currentIndex and index of the remaining element.

There can be 2 edge cases in this.

- What if we found exactly half of the target. Then to confirm we have one more element of the same value, we would check the currentIndex < map.get(target-currentElement). This condition would ensure that we have another element which is ahead of this as map would have already updated the index when we are entering the elements in the array.

Interview problem: First unique Character in String

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

eg. Input: s = "leetcode"

Output: 0

Input: s = "loveleetcode"

Output: 2

Input: s = "aabb"

Output: No Character is found: -1

Code: [LP_code4.java](#)

Output:

```
Enter the String: leetcode
The first non repeating character index is: 0

Enter the String: loveleetcode
The first non repeating character index is: 2

Enter the String: aabb
No character is found: -1
```

Approach:

We would create a hashmap of char, integer. Now for every character of String we would insert this into the hashmap and update the frequency of that element..

Now we would traverse the map and check the first character whose frequency is equal to 1. If we found any such element we found our answer, else we will return -1.

Next Class Teasers

- Stacks In Java
- Queue in Java

