Searching Algorithm, Output = 3

0	1	2	3	4	5	6

arr →

2	4	1	7	10	12	13

target = 7

Random access

arr(3) = 7

Index number
↓
Output = 3

Analysis

Time complexity — $O(n)$
Space complexity — $O(1)$

Linear Search

Psuedocode

```
for (i = 0; i < n; i++) {
    if (arr(i) == target)
    {   print (Element is Present,
                                i);
    }
}
Print (Element is not Present);
```

Advantages

↳ Simple to understand & implement
↳ useful in every scenario of an <u>array</u>

$$\underset{\text{sorted}}{\diagup} \qquad \underset{\text{unsorted}}{\diagdown}$$

↳ No additional memory/space

Disadvantages

n——— very very high (Huge datasets)

↳ <u>Time required is</u> <u>very very high</u>