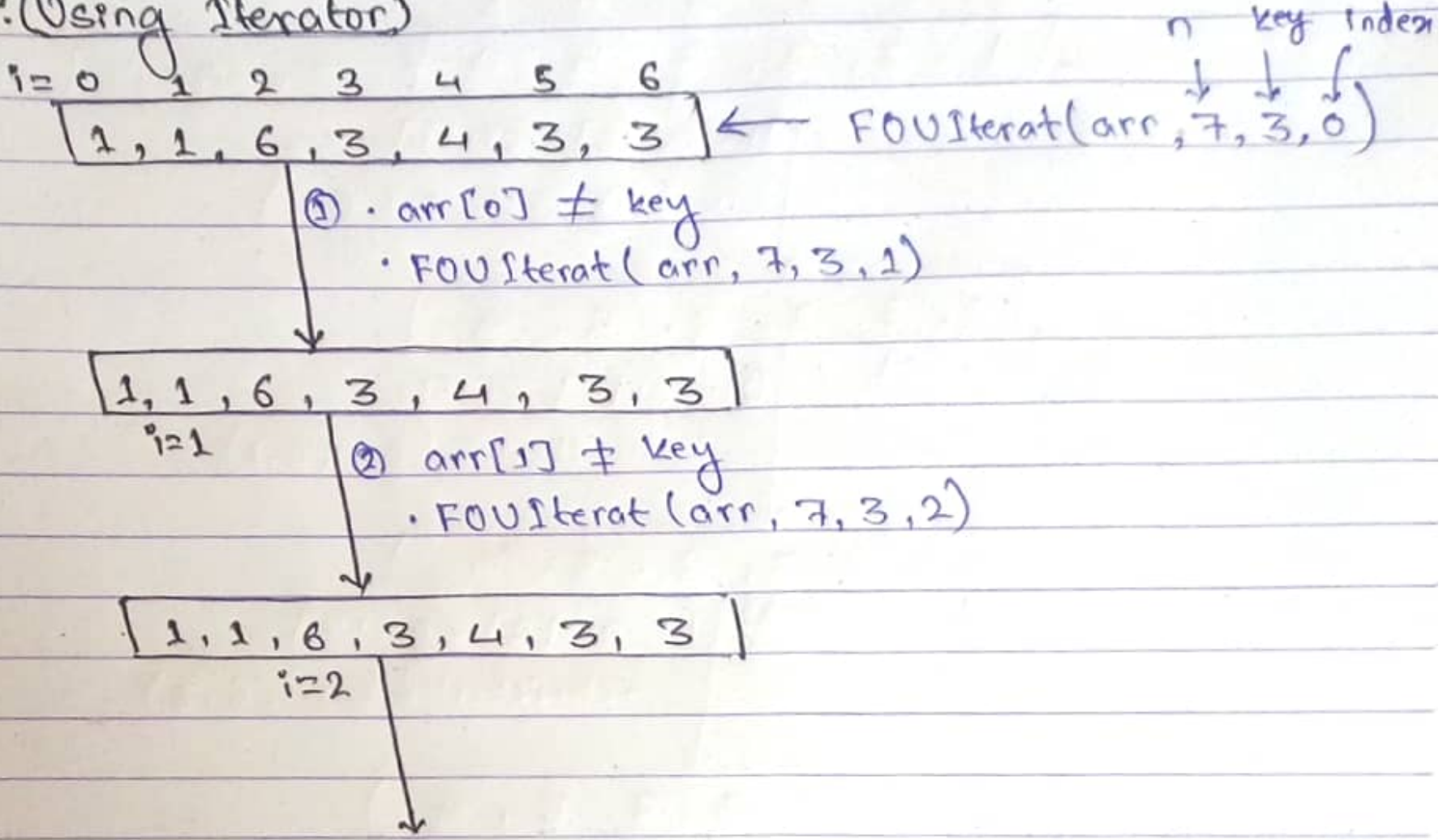
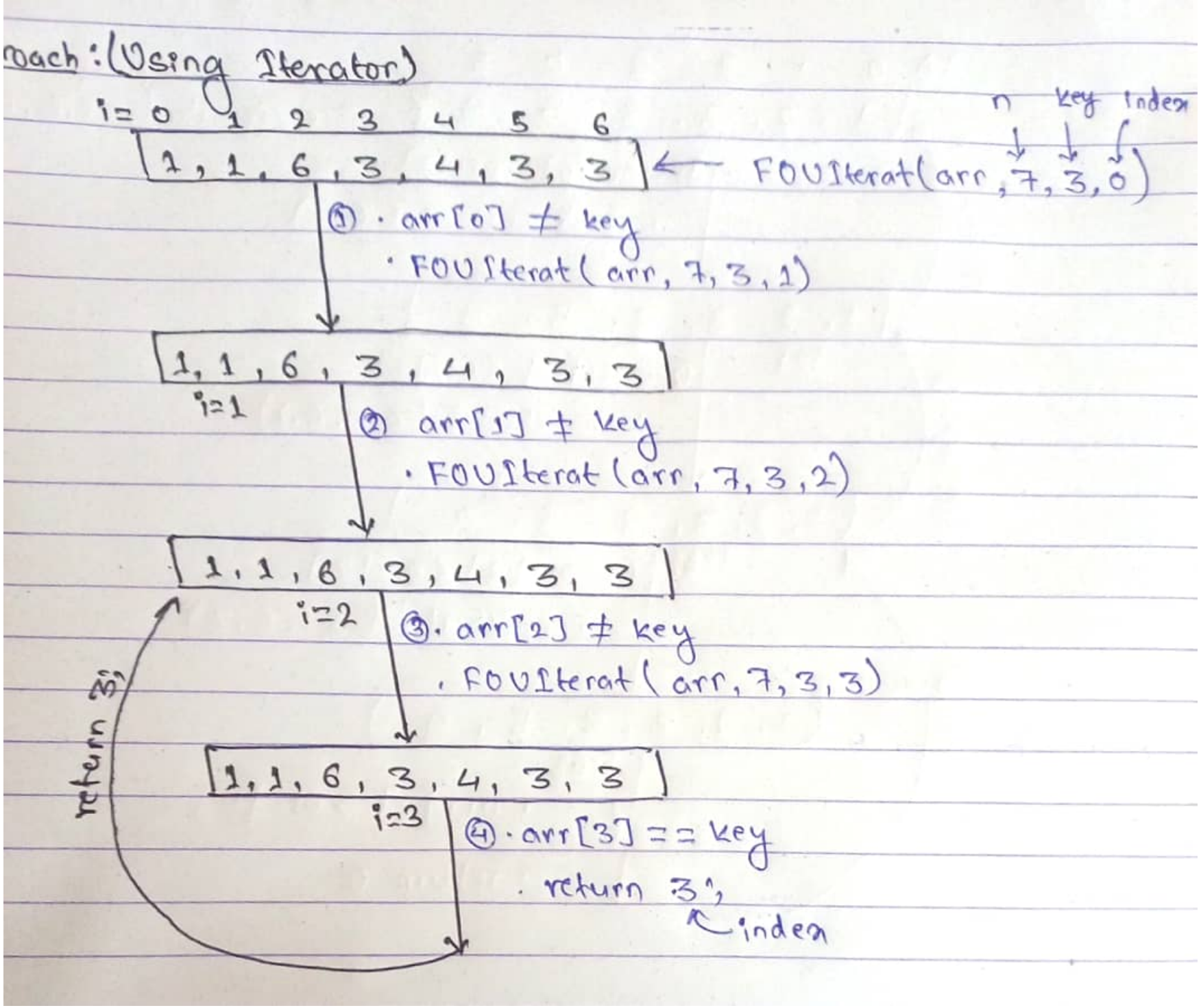
0 1 2 3 4 5 6 7 ← Index

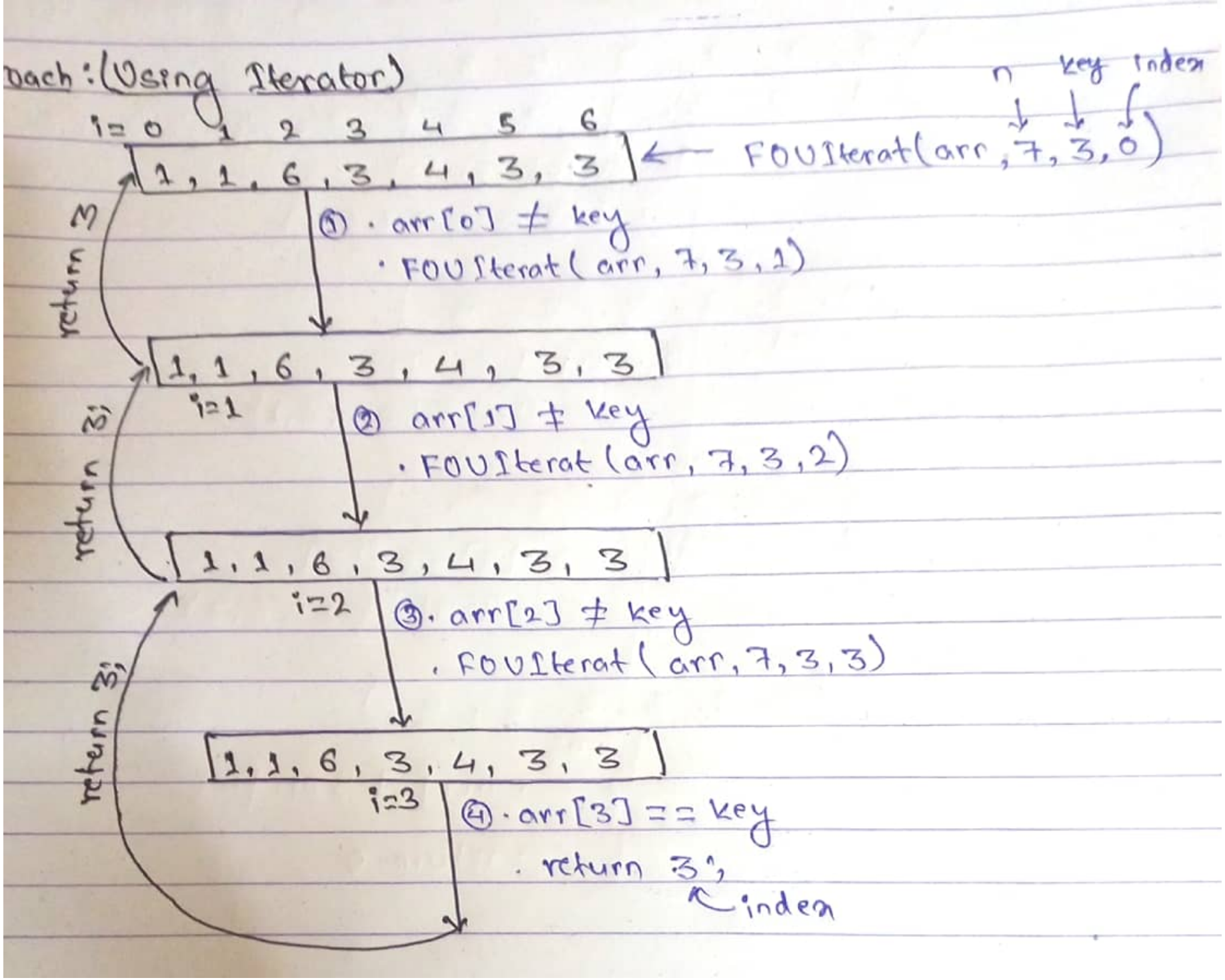
2.) First Occurance ; { 1, 1, 6, 3, 4, 3, 3, 7 };

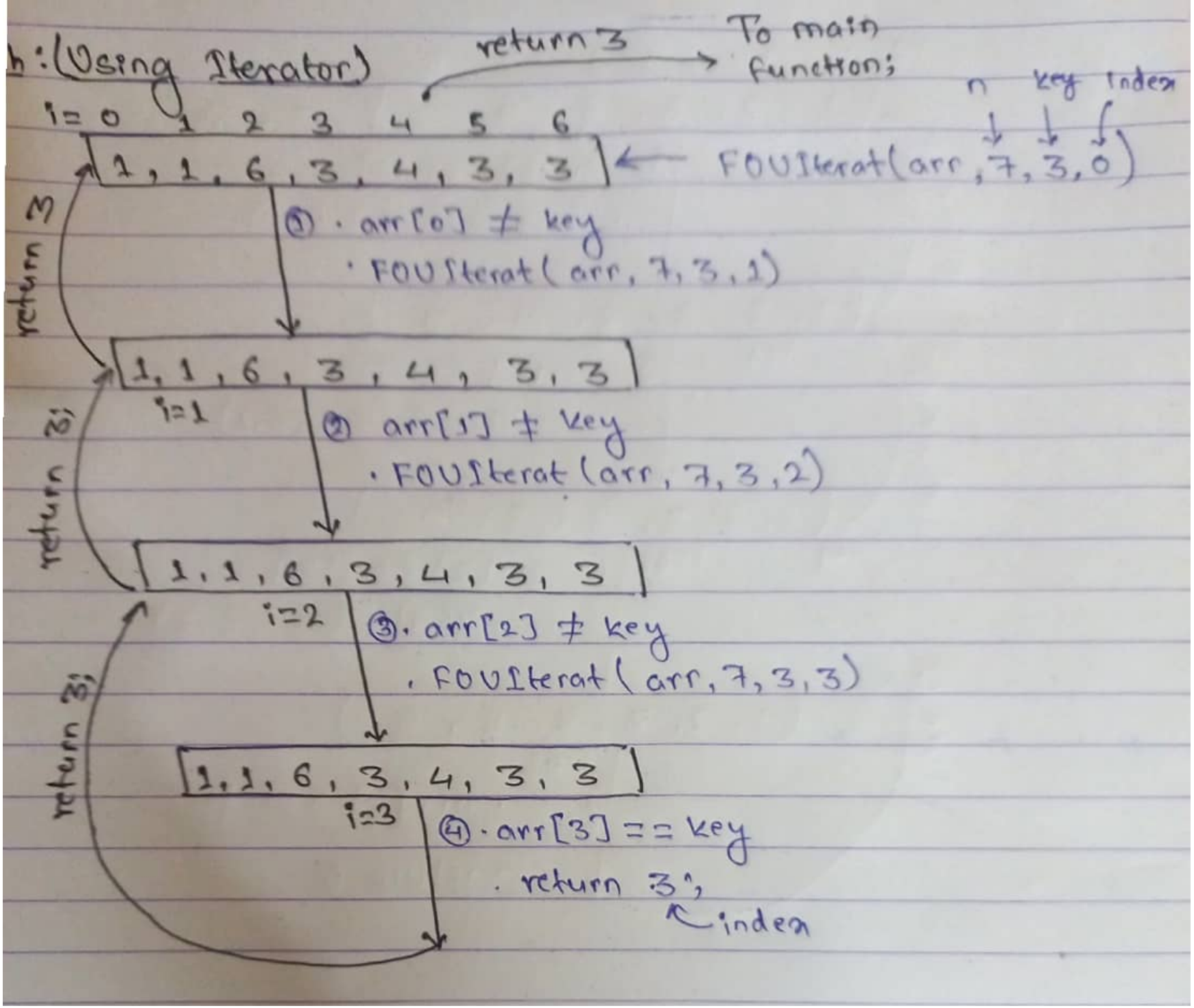
· 1st Occurance of 3 is at index 3.

1st Approach : (Using Iterator)




oach: (Using Iterator)
i= 0 1 2 3 4 5 6
1, 1, 6, 3, 4, 3, 3
n key index
FOUIterat(arr, 7, 3, 0)
① · arr[0] ≠ key
· FOUIterat(arr, 7, 3, 1)
1, 1, 6, 3, 4, 3, 3
i=1
② arr[1] ≠ key
· FOUIterat(arr, 7, 3, 2)
1, 1, 6, 3, 4, 3, 3
i=2
③ · arr[2] ≠ key
· FOUIterat(arr, 7, 3, 3)
return 3;
1, 1, 6, 3, 4, 3, 3
i=3
④ · arr[3] == key
· return 3;
index


Each: (Using Iterator)
n
key
index
i= 0 1 2 3 4 5 6
1, 1, 6, 3, 4, 3, 3
FOUIterat(arr, 7, 3, 0)
① · arr[0] ≠ key
· FOUIterat(arr, 7, 3, 1)
return 3;
1, 1, 6, 3, 4, 3, 3
i=1
② arr[1] ≠ key
· FOUIterat(arr, 7, 3, 2)
return 3;
1, 1, 6, 3, 4, 3, 3
i=2
③ · arr[2] ≠ key
· FOUIterat(arr, 7, 3, 3)
return 3;
1, 1, 6, 3, 4, 3, 3
i=3
④ · arr[3] == key
· return 3;
index

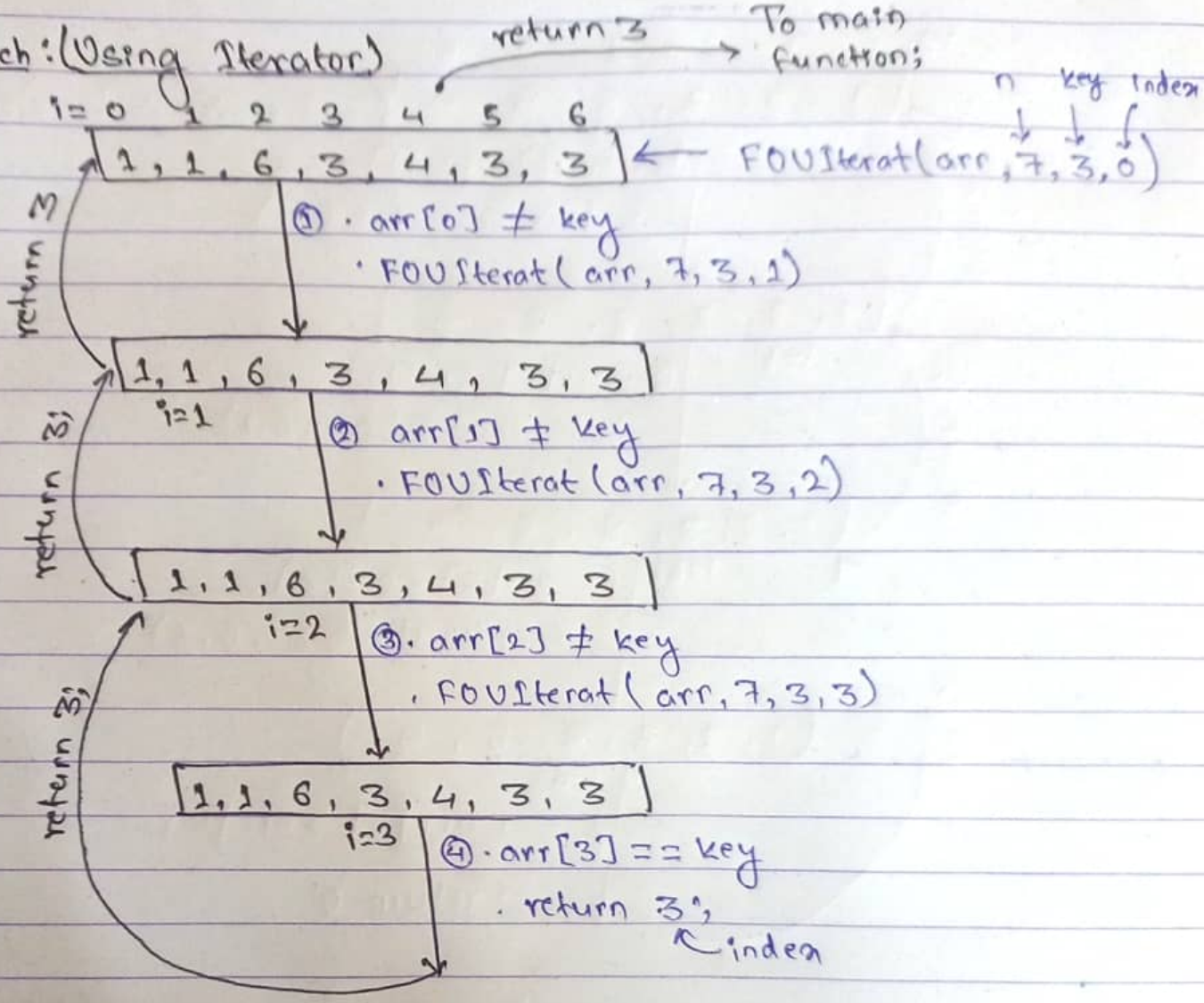

h: (Using Iterator)
return 3
To main function;
i= 0 1 2 3 4 5 6
1, 1, 6, 3, 4, 3, 3
n key index
FOUIterat(arr, 7, 3, 0)
① · arr[0] ≠ key
· FOUIterat(arr, 7, 3, 1)
return 3;
1, 1, 6, 3, 4, 3, 3
i=1
② arr[1] ≠ key
· FOUIterat(arr, 7, 3, 2)
return 3;
1, 1, 6, 3, 4, 3, 3
i=2
③ · arr[2] ≠ key
· FOUIterat(arr, 7, 3, 3)
return 3;
1, 1, 6, 3, 4, 3, 3
i=3
④ · arr[3] == key
· return 3;
index

2.) First Occurance; {1, 1, 6, 3, 4, 3, 3, 7};

0 1 2 3 4 5 6 7 ← Index

- 1st Occurance of 3 is at index 3.

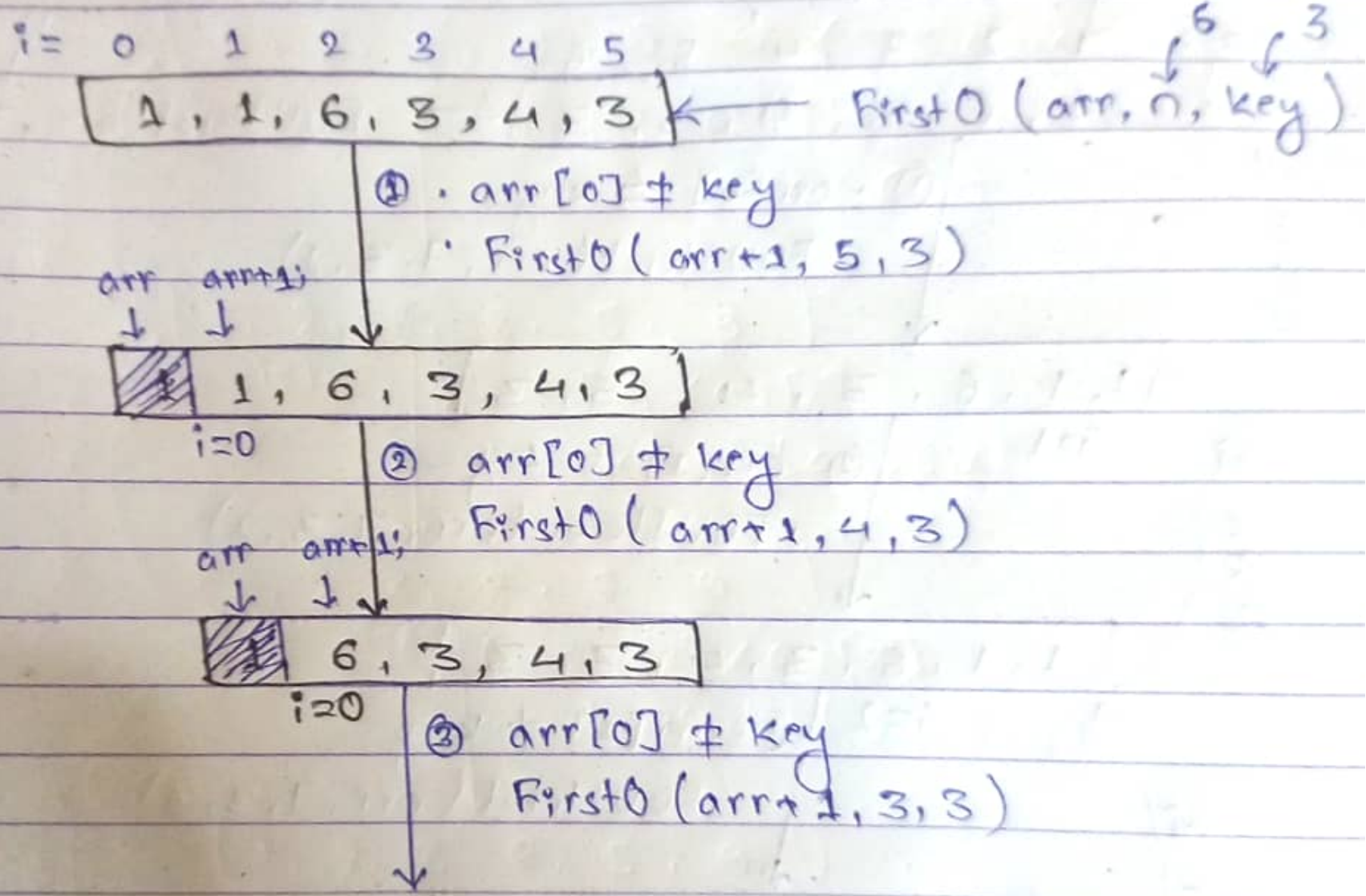
1st Approach: (Using Iterator)



- If completed accessing all element then
  - (index == n)
  - return -1;

i.e (It means Element is not present.)

2nd Approach : Each time Breaking into smaller array;



Approach : Each time Breaking into smaller array;