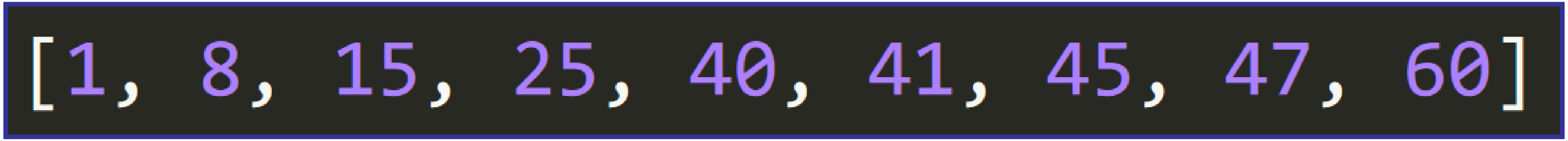
Algorithm

Binary Search

Example

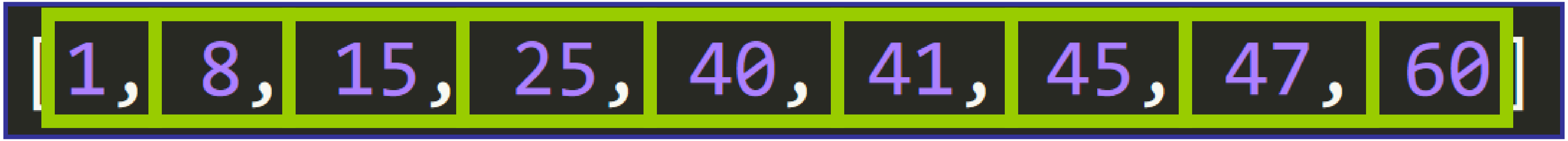
Binary Search

Target: 60



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

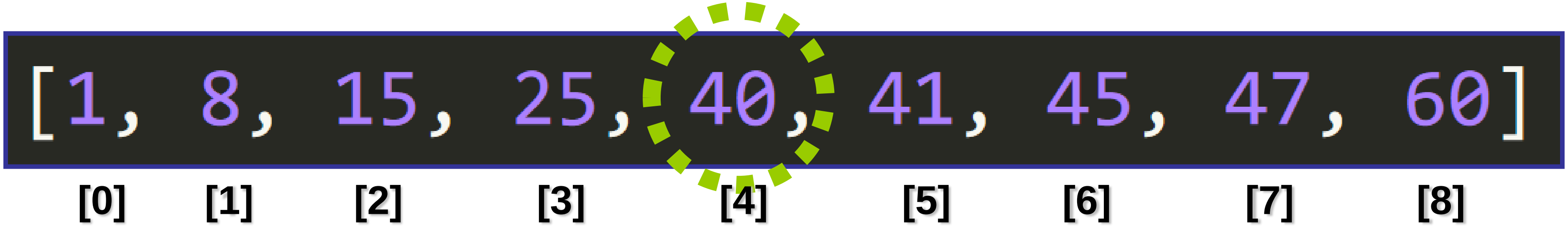
Binary Search



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

[0] [1] [2] [3] [4] [5] [6] [7] [8]

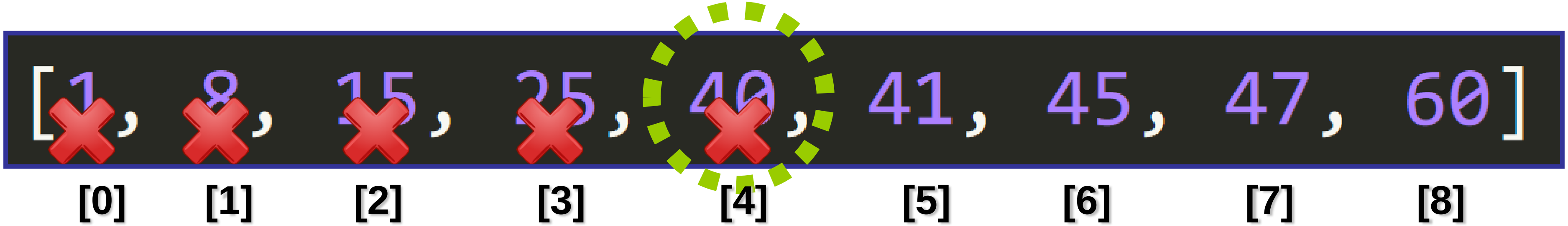
Binary Search



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

Binary Search

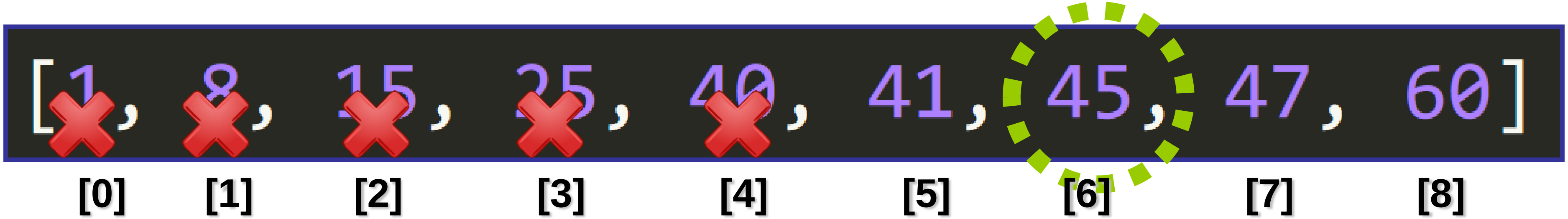
40 < 60



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

[0] [1] [2] [3] [4] [5] [6] [7] [8]

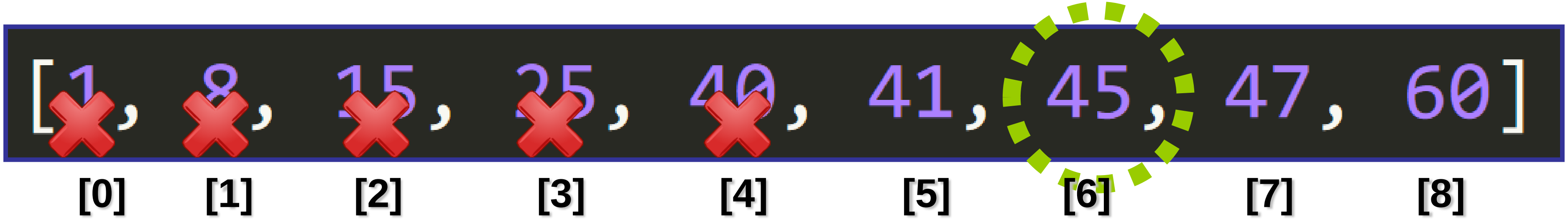
Binary Search



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

Binary Search

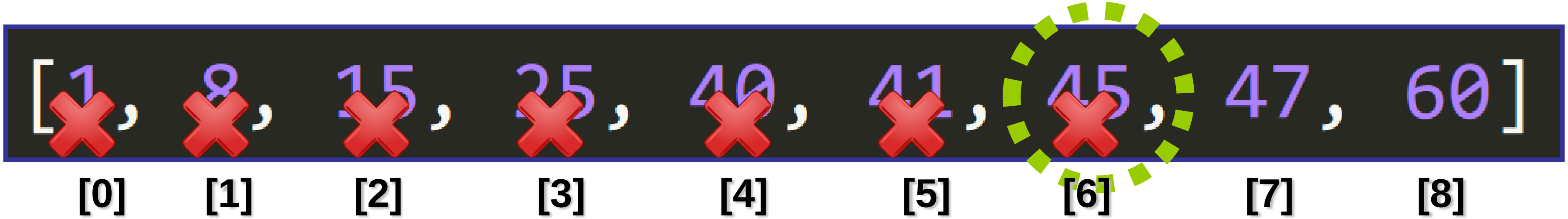
$45 < 60$



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

[0] [1] [2] [3] [4] [5] [6] [7] [8]

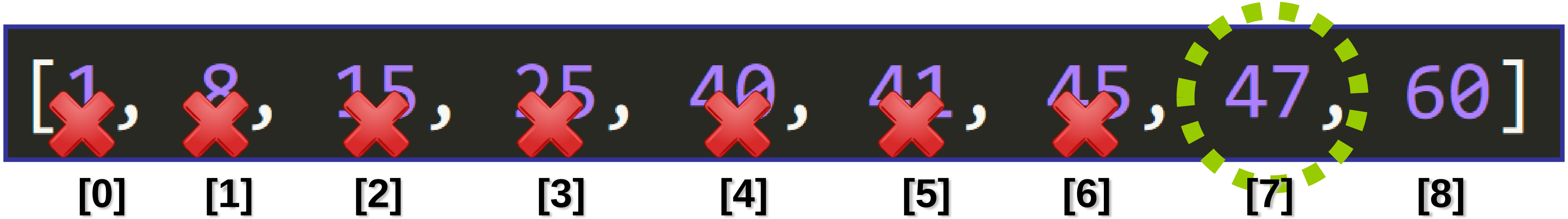
Binary Search



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

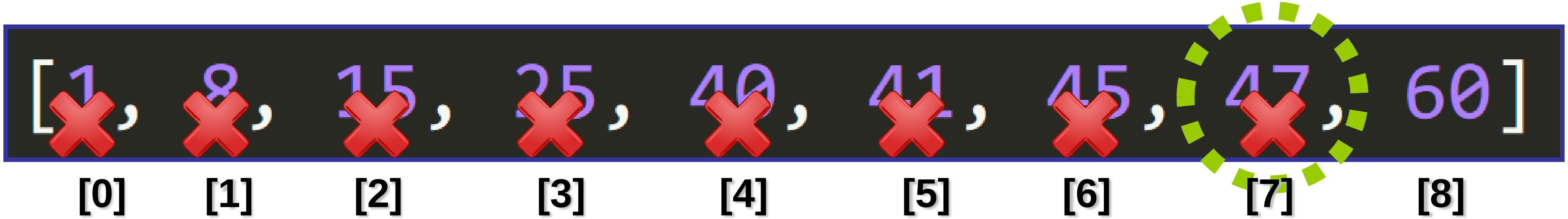
Binary Search

47 < 60



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

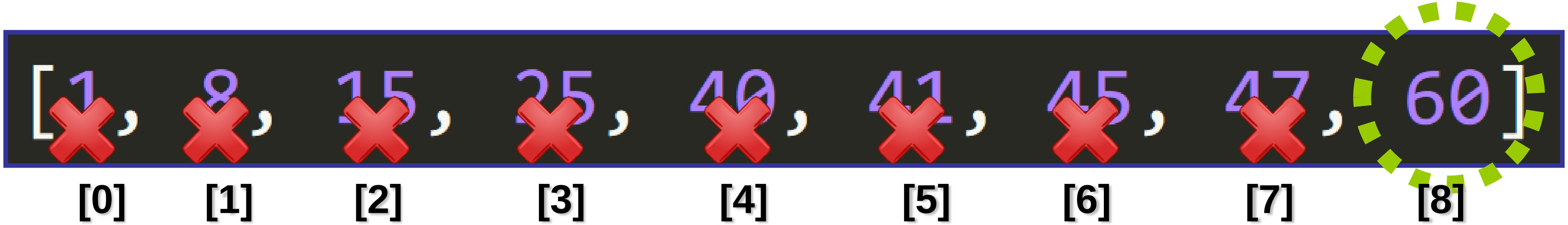
Binary Search



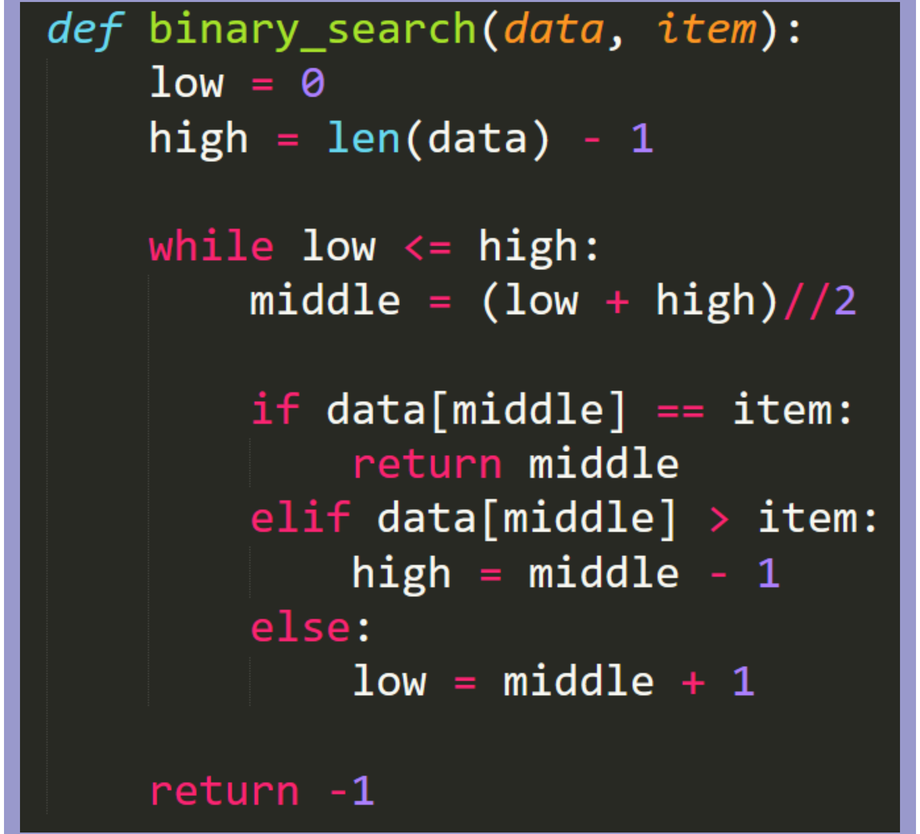
```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

Binary Search

Found at index 8!



```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```



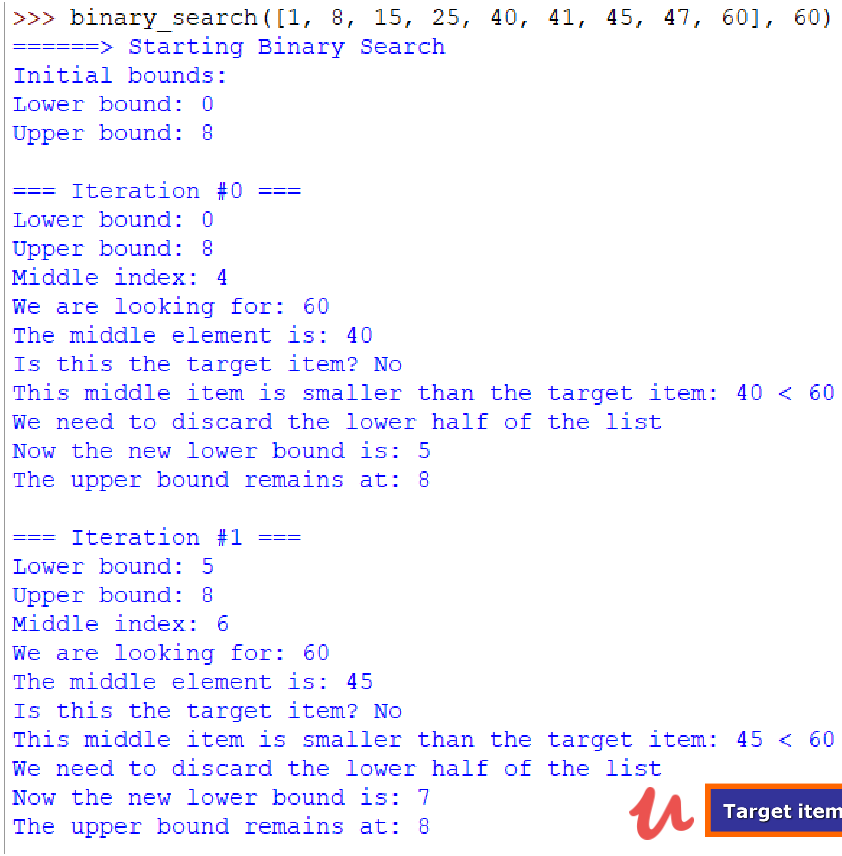
```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```

```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```

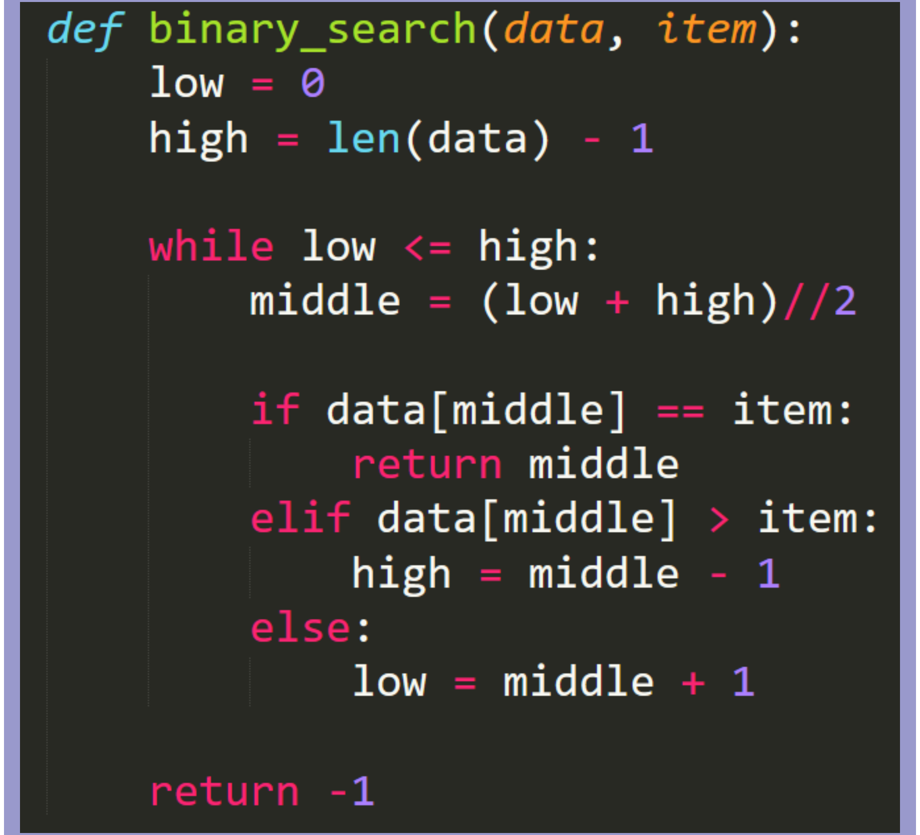


```
>>> binary_search([1, 8, 15, 25, 40, 41, 45, 47, 60], 60)
======> Starting Binary Search
Initial bounds:
Lower bound: 0
Upper bound: 8

=== Iteration #0 ===
Lower bound: 0
Upper bound: 8
Middle index: 4
We are looking for: 60
The middle element is: 40
Is this the target item? No
This middle item is smaller than the target item: 40 < 60
We need to discard the lower half of the list
Now the new lower bound is: 5
The upper bound remains at: 8

=== Iteration #1 ===
Lower bound: 5
Upper bound: 8
Middle index: 6
We are looking for: 60
The middle element is: 45
Is this the target item? No
This middle item is smaller than the target item: 45 < 60
We need to discard the lower half of the list
Now the new lower bound is: 7
The upper bound remains at: 8
```

Target item: 60



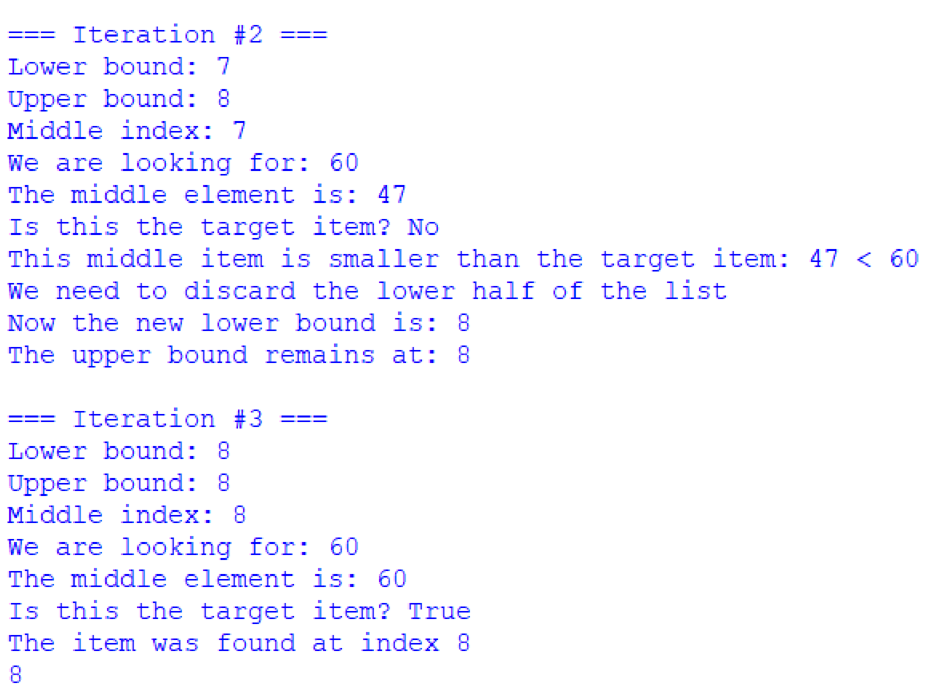
```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```

```
[1, 8, 15, 25, 40, 41, 45, 47, 60]
```



```
=== Iteration #2 ===
Lower bound: 7
Upper bound: 8
Middle index: 7
We are looking for: 60
The middle element is: 47
Is this the target item? No
This middle item is smaller than the target item: 47 < 60
We need to discard the lower half of the list
Now the new lower bound is: 8
The upper bound remains at: 8

=== Iteration #3 ===
Lower bound: 8
Upper bound: 8
Middle index: 8
We are looking for: 60
The middle element is: 60
Is this the target item? True
The item was found at index 8
8
```

Target item: 60

Time to Practice!