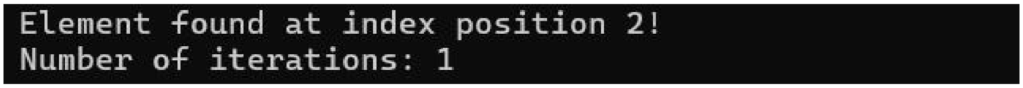
Binary Search

```
class Tester {

  public static int iterations = 0;

  public static int searchElement(int elements[], int low, int high, int
elementToBeSearched) {
    while (low <= high) {
      iterations++; // increment the number of iterations
      int mid = low + (high - low) / 2;

      // Check if elementToBeSearched is present at mid
      if (elements[mid] == elementToBeSearched) {
        return mid; // return the index position
      }

      // If elementToBeSearched is greater, ignore the left half
      if (elements[mid] < elementToBeSearched) {
        low = mid + 1;
      } else {
        // If elementToBeSearched is smaller, ignore the right half
        high = mid - 1;
      }
    }

    return -1; // Element not found
  }


  public static void main(String[] args)
    { int[] elements = { 1, 23, 43, 46, 78,
    90 }; int elementToBeSearched = 43;


    int indexPosition = searchElement(elements, 0, elements.length - 1,
elementToBeSearched);


    if (indexPosition == -1)
      System.out.println("Element not found!");
    else
      System.out.println("Element found at index position " + indexPosition + "!");


    System.out.println("Number of iterations: " + iterations);
  }
}
```



```
Element found at index position 2!
Number of iterations: 1
```