Programming Part

Binary Search (20 pts)

Problem Description

Given an ordered array **a** with N numbers $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$ (possibly repeated), with

$$\mathbf{a}[1] \le \mathbf{a}[2] \le \cdots \le \mathbf{a}[N],$$

run the following binary search on the array to decide whether some *key* is within **a**, as introduced in Lecture 2 of the class.

Note that there are many different variant versions of the binary search algorithm. You are asked to implement the version below *exactly* to produce the correct answer for this problem.

```
BIN-SEARCH($\mathbf{a}$, $key$, $\ell$, $r$)
while $\ell \le r$
    $m$ = floor(($\ell + r$)/2)
    if $\mathbf{a}[m]$ equals $key$
        return $m$
    elseif $\mathbf{a}[m] > key$
        $r = m - 1$ // cut out end
    elseif $\mathbf{a}[m] < key$
        $\ell = m + 1$ // cut out begin
return NIL
```

Input

The first line includes two integers N and *key*, representing the size of the array and the *key* for searching. The second line includes N integers, representing the elements of the array $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$. All numbers are separated by a space.

Output

- For each execution of Line 1 of the binary search algorithm above, print a line of

  ```
  Searching key in range [l, r].
  ```

  with `key` being key, `l` being ℓ and `r` being r.
- Then, if the algorithm returns in Line 4, print a line of

  ```
  Found at index m.
  ```

  with `m` being the m returned.
- Otherwise, if the algorithm returns in Line 9, print a line of

  ```
  Not found.
  ```

Constraint

- $1 \leq N \leq 2^{22}$
- $1 \leq key \leq 2^{22}$
- $1 \leq \mathbf{a}[1] \leq \mathbf{a}[2] \leq \mathbf{a}[3] \leq \cdots \leq \mathbf{a}[N] \leq 2^{22}$

Sample Testcases

Sample Input 1

```
3 4
1 2 4
```

Sample Output 1

```
Searching 4 in range [1, 3].
Searching 4 in range [3, 3].
Found at index 3.
```

Sample Input 2

```
3 3
1 2 4
```

Sample Output 2

```
Searching 3 in range [1, 3].
Searching 3 in range [3, 3].
Searching 3 in range [3, 2].
Not found.
```

Sample Input 3

```
4 64
1 2 89 1126
```

Sample Output 3

```
Searching 64 in range [1, 4].
Searching 64 in range [3, 4].
Searching 64 in range [3, 2].
Not found.
```

Hint

- By design, you can pass this homework by simulating the binary search algorithm properly. There is no need for other arithmetic calculations or cuts.