Ans. to Q.No-1

1) The equation ~~will~~ for finding remainder will be:

[∵ Remainder = Dividend - (Divisor × quotient)]

$$A - \left(B * \text{floor}\left(A/B \right) \right) .$$

2)

Operator	Count
−	1
*	1
/	1

Ans to Q No - 2

⇒ First, we find the lower bound of the given integer in the array and then find the upper bound. After that, (index of upper bound - index of lower bound) will give us the results.

⇒ We can also do it using,

Binary search first:

```
int idx = binary_search(a, a+n, p) - a;

int count = 1;
```

Code for counting elements on left side:

```
int l = idx-1;
while ( l>=0 && a[l]==p) {
    ++count;
    --l;
}
```

For counting right side:

```
int r = idx+1;
while (r<n && a[r]==p) {
    ++count; ++r;
}
```

count is the answer / no of occurences

Ans. to QNo-3

1. Array
2. Linked list
3. Double ended Queue
4. Binary Search Tree

Ans to Q No - 4

We will take 2 balls each time and weigh them. When we will find an imbalance we will take a ball from that imbalance pair and weigh it against another ball. This will generate the solution. In this approach, we will need to use the scale maximum 5 times.

Ans to Q No - 5

We may use a Stack to solve in O(N).

We iterate through the array and insert the current number in the stack only if the stack is empty or s.top() is a number > current number.

But when current number is greater than s.top(), we pop all the elements currently present in the stack and print them because the current number is greater than all those numbers. Only after that, we put the current number in the stack and repeat the process. When iteration is done, we print the numbers left in the stack as number -> none.