

Problem #8

We can use binary search because all toys are produced in order with a special serial.

Complexity {
 → best case $O(1)$
 → average case $O(\log n)$
 → worst case $O(\log n)$

Steps :-

First we start searching from the middle
 $\text{int mid} = \text{Number of toys} / 2;$

Then we start to check the middle toy
if we find the toy is defective
we find the defective toys in order (1)

if the middle toy is not defective we
can search in the second half of toys
 $\text{int low} = \text{mid} + 1;$
 $\text{int high} = \text{Number of Toys};$

if The middle Toys is defective we can search in first half of toys.

int low = 1;

int high = mid - 1;

C++ Code

```
int Binary-Search(int *arr, int low, int high, int target)
while (low ≤ high)
{
    int mid = [low + (high - 1)] / 2
    if (arr[mid] == target)
        return mid;
    if (arr[mid] > target)
        return Binary-Search(arr, low, mid - 1, target);
    return Binary-Search(arr, mid + 1, high, target);
}
```