MINIMUM DAYS TO MAKE M BOUQUETS

☆ In this problem, we are given an array of integers of size N which contains the date on which i^{th} flower will bloom. We are also given two integers m, K and we are supposed to return the minimum number of days it would take to make m bouquets with K adjacent flowers within them.

Eg :	[7	7	7	7	13	11	12	7]
On 7th Day	✓	✓	✓	✓	×	×	×	✓
On 12th Day	✓	✓	✓	✓	×	✓	✓	✓

If m * K > N, we return -1 as getting m bouquets of K flowers each is not possible.

Brute force solution is to iterate through each element from minim to maximum, then at each step we check if we got adjacent K flowers that bloomed.

In the optimal solution, we can turn this into a binary search

Pseudocode

```
possibility (arr, N, m, K, day) {
    c = 0, n = 0
    for (i = 0 → N) {
        if (arr[i] <= day) {
            c++
        } else {
            n += c/K
            c = 0
        }
    }
    n += c/K
    if (n > m) return true
    else return false
}
minimumDays(arr, N, m, K) {
    mini, maxi = 0, 0
    for (i = 0 → N) {
        if (arr[i] < mini) {
            mini = arr[i]
        }
        if (arr[i] > maxi) {
            maxi = arr[i]
        }
    }
    minDays = INT_MIN
    while (mini <= maxi) {
        mid = (mini + maxi) |2
        days = possibility (arr, N, M, K, mid)
        if(days == true && mid < minDays){
            minDays = mid
            maxi = mid - 1
        } else {
            mini = mid + 1
        }
    }
    return minDays
}
```