

12th June 2025

Nicole Page No. _____
Date: _____

Problem 1: Flipper

Approach:

We check for task completion day by day and keep track of which tasks are already done using a boolean [] done. For each day:

1. Mark task A[day] as done.
2. Check:

- if $A[\text{day}] - k - 1 \geq 1$, and that task is done and all in b/w are not.
3. If $A[\text{day}] + k + 1 \leq N$ and that task is done and all in b/w are not.

Pseudo-code:

```
for (int day = 0; day < N; day++) {
    task = A[day];
    done[task] = true;
    if (task - k - 1 >= 1 && done[task - k - 1]) {
        boolean allNotDone = true;
        for (int i = task - k; i < task; i++) {
            if (!done[i]) {
                allNotDone = false;
                break;
            }
        }
        if (!allNotDone) return day + 1;
    }
    if (task + k + 1 <= N && !done[task + k + 1]) {
        boolean allNotDone = true;
        for (int i = task + 1; i < task + k + 1; i++) {
            if (!done[i]) {
                allNotDone = false;
                break;
            }
        }
        if (!allNotDone) return day + 1;
    }
}
```

say run:-

9/P : 2

2, 1

1, 1

4, 1

O/P : - 1 2 4 3
3

=> Test 1: Not enough distinct task \rightarrow impossible \rightarrow 1

Test 2: Day 1: complete task 1 \rightarrow done = {1}

Day 2: " " 2 \rightarrow done = {1, 2}

Day 3: " " 4 \rightarrow now task 1 and 4 are done, with task 2 and 3 in b/w.

- check task 1 and 4 \rightarrow 2 uncompleted b/w \rightarrow 2 = 2 k?

NO,

- But task 2 and 4 \rightarrow 1

uncompleted (3) b/w \rightarrow 1 = 3 \rightarrow returns

Problem 2:

Approach:- calculate total sum of array
 If sum is odd, return sum
 If sum is even:
 • find the smallest odd no.
 • subs it from sum to make it odd.
 • return the new sum
 If there is no odd no. → return -1

pseudo-code:

```

int sum = 0;
int minOdd = Integer.MAX_VALUE;
boolean hasOdd = false;
for (int num : A) {
    sum += num;
    if (num % 2 != 0) {
        hasOdd = true;
        minOdd = Math.min(minOdd, num);
    }
}
if (sum % 2 == 0) return sum;
if (!hasOdd) return -1;
return sum - minOdd;
    
```

Program; gfp: 4 2 6 3 2

$$\Rightarrow \text{sum} = 4 + 2 + 6 + 3 + 2 = 17$$

17 is odd → return 17.