Task 5 — Sorting Light and Dark Disks

1. Problem Description

We are given 2n disks arranged in an alternating pattern:

- D = Dark disk
- L = Light disk

Initial arrangement:

D L D L D L

The objective is to sort the disks so that:

- All light disks (L) are positioned on the left
- All dark disks (D) are positioned on the right

2. Approach Used

I implemented a bubble-sort style algorithm.

Steps:

- Create an array of size 2n.
- Fill it with alternating 'D' and 'L'.
- Repeatedly scan the array from left to right.
- Keep repeating passes until no more swaps are needed.
- Count how many swaps were performed.

This method guarantees that eventually all L moves are left and all D moves are right.

3. Code Explanation

a. function

```
vector<char> disks(2 * n);

for (int i = 0; i < 2 * n; i++) {

    disks[i] = (i % 2 == 0 ? 'D' : 'L');

}
```

This code creates the initial alternating pattern of disks. It builds a vector of size 2n, then fills each position based on whether the index is even or odd. Even indices receive a dark disk 'D', and odd indices receive a light disk L. As the loop progresses, this automatically produces the sequence D L D L D L, which is the required starting arrangement for the sorting algorithm.

b. Sorting the disks

```
while (!sorted) {

    sorted = true;

    for (int i = 0; i < 2*n - 1; i++) {

        if (disks[i] == 'D' && disks[i + 1] == 'L') {

            swap(disks[i], disks[i + 1]);

            count++;

            sorted = false;

        }

    }
}
```

Here, we check each pair of neighbors and swap only if they are in the same direction (D) and in the same level (L). Continue until the entire array is sorted. And the count keeps track of moves.

4. Output Example

For n = 3:

Final arrangement: L L L D D D

Total moves: 9

5. Time Complexity

The algorithm behaves like bubble sort:

- $O(n^2)$ comparisons
- $O(n^2)$ swaps

6. Conclusion

- Successfully sorts alternating disks.
- Accurately counts swap operations.
- Simple and easy to implement.
- Not the most efficient, but guarantees correctness.