

Coding PYQ-Style Set

These are public-pattern coding questions repeatedly seen in placement prep. They are not guaranteed exact exam questions.

Set A: Easy high-frequency

1. Sum of array

Input:

```
5
1 2 3 4 5
```

Output:

```
15
```

Core logic:

```
long long sum = 0;
for (int x : a) sum += x;
```

2. Largest and smallest element

Output both largest and smallest.

Core logic:

```
int mn = a[0], mx = a[0];
for (int x : a) {
    mn = min(mn, x);
    mx = max(mx, x);
}
```

3. Count even and odd

Core logic:

```
if (x % 2 == 0) even++;
else odd++;
```

4. Palindrome number

Reverse number and compare with original.

5. Prime in range

Use `isPrime()` for each number from L to R.

Set B: Medium

6. Second largest distinct

Handle duplicates. If all same, print no second largest or `-1`, depending on question.

7. Remove duplicates from array

If order matters:

```
unordered_set<int> seen;
vector<int> result;
for (int x : a) {
    if (!seen.count(x)) {
        seen.insert(x);
        result.push_back(x);
    }
}
```

8. Find missing number from 1 to n

If array has n-1 numbers:

```
expected = n * (n + 1) / 2
missing = expected - actualSum
```

Use `long long`.

9. Check anagram

Use frequency count.

10. First non-repeating character

Count frequency, then scan original string.

Set C: Higher-value

11. Pair sum

Use hash set or sorting + two pointers.

12. Maximum subarray sum

Use Kadane's algorithm.

13. Rotate array by k

Normalize:

```
k = k % n;
```

For right rotation:

```
reverse(a.begin(), a.end());  
reverse(a.begin(), a.begin() + k);  
reverse(a.begin() + k, a.end());
```

14. Merge two sorted arrays

Use two pointers.

15. Character frequency

Use array of size 256 or map.

Last-day coding drill

Implement without looking:

- prime check,
- palindrome string,
- second largest,
- frequency map,
- GCD,
- reverse number,
- anagram,
- pair sum.

If you can code these cleanly, you are ready for most easy/medium placement coding questions.