

01. Coding Strategy & Language Boilerplates

1. CoCubes Coding Round Environment Strategy

- **Duration:** 30 Minutes for 2 Questions.
- **Languages Supported:** C, C++, Java, Python.
- **Key Success Factor:** Passing 100% of hidden test cases (including edge cases: negative inputs, single element arrays, empty strings, integer overflow).

Edge Cases & Test Case Checklist

1. **Empty / Minimal Input:** Arrays of size 0 or 1, empty strings, single character strings.
 2. **Extreme Values:** Minimum and maximum integer bounds (`INT_MIN` , `INT_MAX`), potential integer overflow when multiplying or accumulating sums.
 3. **Duplicates:** Arrays containing all identical elements, duplicate keys, or all non-unique values.
 4. **Sign Extremes:** Negative numbers, zero, all negative arrays (e.g. Kadane's algorithm).
 5. **Character Cases:** Mixed case strings, special characters, spaces, or numeric digits in string inputs.
-

2. Fast I/O Boilerplates

C Boilerplate

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <string.h>

void solve() {
    int n;
    if (scanf("%d", &n) != 1) return;
    // Process input
}

int main() {
    solve();
    return 0;
}
```

C++ Boilerplate

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
```

```
#include <climits>

using namespace std;

void solve() {
    int n;
    if (!(cin >> n)) return;
    // Process input logic here
}

int main() {
    // Fast I/O optimization
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    solve();
    return 0;
}
```

Java Boilerplate

```
import java.util.*;
import java.io.*;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = null;

        String line = br.readLine();
        if (line == null || line.trim().isEmpty()) return;
        st = new StringTokenizer(line);

        // Read input logic here
    }
}
```

Python Boilerplate

```
import sys

def solve():
    input_data = sys.stdin.read().split()
    if not input_data:
        return
    # Process input logic here
```

```
if __name__ == '__main__':  
    solve()
```