C++ Competitive Programming Cheatsheet

C++ Competitive Programming Cheatsheet

Case 1: Single test case, single input, single output

Boilerplate:
```
#include <bits/stdc++.h>
using namespace std;

void solve() {
    // Write your solution here
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}
```

Case 2: Single test case, multiple inputs until EOF

Boilerplate:
```
#include <bits/stdc++.h>
using namespace std;

void solve() {
    int x;
    while (cin >> x) {
        // Process each input
    }
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}
```

Case 3: Multiple test cases with t given

Boilerplate:
```
#include <bits/stdc++.h>
using namespace std;

void solve() {
```

```
    int t;
    cin >> t;
    while (t--) {
        // Solve for each test case
    }
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}
```

Case 4: Single test case, multiple lines of input

Boilerplate:

```
#include <bits/stdc++.h>
using namespace std;

void solve() {
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        string s;
        cin >> s;
        if (s.length() > 10) {
            cout << s.front() << s.length() - 2 << s.back() << "\n";
        } else {
            cout << s << "\n";
        }
    }
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}
```

Case 5: Grid or matrix input

Boilerplate:

```
#include <bits/stdc++.h>
using namespace std;

void solve() {
```

```
    int n, m;
    cin >> n >> m;
    vector<vector<int>> grid(n, vector<int>(m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> grid[i][j];
        }
    }
    // Process grid here
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}

-----------------------------------------------------

Case 6: String manipulation problem
-----------------------------------
Boilerplate:
#include <bits/stdc++.h>
using namespace std;

void solve() {
    string s;
    cin >> s;
    // Process string
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    solve();
    return 0;
}
```