Alright — here’s your **exclusive C++ Codeforces input parsing cheat sheet** with:

* **All 7 cases** from earlier
* **Fast I/O boilerplate** (#include <bits/stdc++.h>, ios::sync\_with\_stdio(false); cin.tie(NULL);)
* **"// --- Solve here ---"** section so you know where to write logic
* Each case is **ready-to-copy** and minimal edits needed for the problem.

**📜 C++ Codeforces Input Parsing Cheat Sheet (with Boilerplate)**

**Case 1 — Single integer**

**Example input:**

n

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

int n;

cin >> n;

// --- Solve here ---

cout << n << "\n";

return 0;

}

**Case 2 — Multiple integers in one line (fixed count)**

**Example input:**

a b c

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

int a, b, c;

cin >> a >> b >> c;

// --- Solve here ---

cout << a + b + c << "\n";

return 0;

}

**Case 3 — First line n, second line array of size n**

**Example input:**

5

1 2 3 4 5

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

int n;

cin >> n;

vector<int> arr(n);

for (int i = 0; i < n; i++) cin >> arr[i];

// --- Solve here ---

int sum = 0;

for (int x : arr) sum += x;

cout << sum << "\n";

return 0;

}

**Case 4 — Multiple test cases, each with simple inputs**

**Example input:**

3

4

5

6

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

// --- Solve here ---

cout << n \* n << "\n";

}

return 0;

}

**Case 5 — Multiple test cases, each with n and then an array**

**Example input:**

2

3

1 2 3

4

4 5 6 7

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

vector<int> arr(n);

for (int i = 0; i < n; i++) cin >> arr[i];

// --- Solve here ---

int sum = accumulate(arr.begin(), arr.end(), 0);

cout << sum << "\n";

}

return 0;

}

**Case 6 — Grid / Matrix input**

**Example input:**

3 4

1 2 3 4

5 6 7 8

9 10 11 12

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

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];

// --- Solve here ---

int total = 0;

for (auto &row : grid)

for (int val : row)

total += val;

cout << total << "\n";

return 0;

}

**Case 7 — All numbers in one line / entire file**

**Example input (one line):**

1 2 3 4 5 6

**Example input (multiple lines):**

1 2

3 4

5 6

#include <bits/stdc++.h>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

vector<int> data;

int x;

while (cin >> x) {

data.push\_back(x);

}

// --- Solve here ---

int sum = accumulate(data.begin(), data.end(), 0);

cout << sum << "\n";

return 0;

}

✅ **How to use this cheat sheet:**

1. Look at the **input format** in the problem.
2. Match it to the case number above.
3. Copy that block, paste into your file, and fill in the // --- Solve here --- part.