

This short guide explains how to write solutions to CEOI 2019 problems in C++.

About your program:

- The program must be a console application (i.e., no GUI).
- The program must read stdin and write to stdout, unless instructed otherwise in the statement.
- The grader discards whatever you print to stderr.
Printing to stderr is allowed, but it may slow your solution down.
- When compiling your program locally, we recommend using the same switches as the grader:
`-static -std=gnu++17 -O2 -Wall -Wextra`

About the input:

- Input files have Linux newlines (only LF, ASCII 10, not CR+LF).

About the output:

- Each line of your output **must be terminated by a newline**.
- Follow the exact output specification from the statement.
Do not print extra whitespace you were not instructed to print.

About large I/O:

- If the input is large, it is recommended to either use `scanf`, or to use `cin`, but toggle off synchronization between C and C++ streams and turn off flushing `cout` before reading from `cin`. In order to do this, you can use `ios_base::sync_with_stdio(false);` and `cin.tie(nullptr);`

Example:

A correct program that inputs two integers and outputs their sum.

Note that it also prints a newline after the answer.

```
#include <iostream>
using namespace std;

int main() {
    int a,b;
    cin >> a >> b;
    cout << (a+b) << endl;
}
```

Using and printing floating-point numbers:

Use the data type `double` to store floating-point numbers.

Use `printf("%.15f", x);` to print the contents of `double x` to 15 decimal places.

Alternately, you can `#include <iomanip>` and then execute the command

`cout << fixed << setprecision(15);` at the beginning of your program. Once you do that, normal printing to `cout` will print `double` variables to 15 decimal places.

In the **scissors** task, it is recommended to print all numbers to at least 10 decimal places.