C++ Basic Configuration for Competitive Programming - CPCFI

Build a basic C++ template

- Start with a C++ file
- Sync the file with a snippet (https://snippet-generator.app/)

IO Code

C++ Fast Input Output

- Using `cin` and `cout` in C++ is slower than `scanf` and `printf` since both previous funcitons need to keep in sync with the underlying C library which allows using C and C++ IO functions in the same program
- There are two main alternatives to this problem, first:
  - Using `ios_base::sync_with_stdio(false);` which toggles on or off the synchronization of all the C++ standard streams with their corresponding C streams
  - And using `cin.tie(0);` which is a method that guarantees the flusing of `cout` before `cin` accepts an input
- Second:
  - Using `cin.tie(0)` and `cout.tie(0)`
- In addition to both of these solutions, it is not recommended to use `cout << endl` since `endl` by default is slower because it forces a flushing stream
- Concretely, `endl` is a stream manipulator that combines writing a newline with flushing the stream.
  - 1) `os << "Hello\n" << std::flush;`
  - 2) `os << "Hello" << std::endl;`
- 1) and 2) are exactly the same. `std::flush` is used to force all buffered writes to take place immediately.

Reading and writing files

- Get current file name using `string file= __ FILE __` macro
- Remove current file extension with `file = string(file.begin(), file.end()-3`
- Generate input file
  - File name: `string input_file = file + "in"`
  - Reopen input stream: `freopen(input_file.c_str(), "r", stdin)`
- Generate output file
  - File name: `string output_file = file + "out"`
  - Reopen output stream: `freopen(output_file.c_str(), "w", stdout)`
- Wrap all these points inside a function, for example:
  - Function: `void setIO() { … }`
  - Call that function before coding your solution: `setIO()` and you are ready to start reading inputs from file `file.in` and writing your solution to `file.out`

Configuring Environment Variable

- The problem with `setIO()` resides in the fact that most OJ (online judges) don't expect you to write your solution to a specific file (there are some OJs that will ask you to read from a file and write your solution to a file)
- In order to avoid this problem, we can define an environment variable that will run `setIO()` only when the environment variable is present in the computer
- **Unix based computers:**
  - Open the shell configuration file (usually `bash.rc`): `nano ~/.bashrc`
  - Define your environment variable name (maybe `CP_IO`) by writing `export CP_IO=true` anywhere on the shell's configuration file
  - Save the configuration file
- **Windows based computers:**
  - Right click on the Windows logo or type `win key + x`
  - Select System > Advanced system settings
  - Advanced tab > Environment Variables (button in the lower right of the screen)
  - Click on `New` button and define the variable name (`CP_IO`) and variable value (`true`)
- Once `CP_IO` variable is defined, let's go back to C++
  - We'll only call `setIO()` if `CP_IO` exists. To do this, we can write the following: `if (getenv("CP_IO"))` and if it evaluates to `true` we'll call `setIO()`

CodeRunner extension

- Open extensions tab in VSCode (Ctrl + Shift + X)
- Install **Code Runner** extension by Jun Han
- Open VSCode shortcuts by pressing `cmd + Shift + P` or `Ctrl + Shift + P`
- Type `shortcuts` and open the option **Preferences: Open Keyboard Shortcuts**
- Search for **run code** and type your desired keybinding (I use `cmd + '` ← single quote)
- Back in C++ file, open again VSCode settings: `cmd + Shift + P` , type **settings** adn open **Preferences: Open Settings (JSON)**
- In the `settings.json` file search for `code-runner.executorMap` and type the following in the **cpp** option
  - Write: `cd $dir && g++-11 $fileName -o $fileNameWithoutExt && $dir$fileNameWithoutExt`
  - This will compile and run your C++ code automatically when you press the keybinding
- Back in the C++ file, run your code using your previously defined keybinding

References

- Fast I/O for Competitive Programming - GeeksforGeeks
  A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and pro...
  https://www.geeksforgeeks.org/fast-io-for-competitive-programming/

- **cin / scanf comparison**
  cin / scanf comparison. GitHub Gist: instantly share code, notes, and snippets.
  https://gist.github.com/tobin/3845568
- **C++ Tutorial => Flushing a stream**
  Learn C++ - Flushing a stream
  https://riptutorial.com/cplusplus/example/6708/flushing-a-stream
- **std::ios_base::sync_with_stdio - cppreference.com**
  https://en.cppreference.com/w/cpp/io/ios_base/sync_with_stdio
  https://en.cppreference.com/w/cpp/io/ios_base/sync_with_stdio