

# Instructions

---

Your coursework will be in part examined through automated testing, so make sure to follow these instructions carefully and to be mindful of formatting.

1. Download the provided files, and ensure that it compiles properly using the `check.sh` script.

```
./check.sh
```

(on Windows, you should be able to run `g++ -o run main.cpp ../submission/submission.cpp` `-I ../submission/` directly in the command prompt)

2. The script outputs a `run` executable which should tell you which of the provided tests succeed and which fail (initially, all but the `empty` tests should fail.)
3. Inside the `submission.cpp` file, implement the necessary functionality of the `asmToCode` and `codeToRegisters` functions. The tests provided do not cover all test-cases, and you are expected to write your own tests in addition to the ones provided which may test more nuanced aspects of your implementation.
4. Feel free to add extra files and functions inside the submission folder, but always ensure that your final submission can properly compile using the original `check.sh` script.
5. Once you have confirmed that your code is compiling properly, and working according to coursework specifications, submit your code for grading.  
*Only submit code from the submission folder. Do not include the `main.cpp` or `main.h` files in your code submission, but you are expected to modified them for the purpose of testing. Therefore, do speak of it in your report and include them as an appendix.*

Note: Do not modify the provided `check.sh` script or the `asmToCode` and `codeToRegisters` function declarations in the `submission.h` header file. Also, make sure the format of the printed messages matches the format seen in the tests subfolders. remember that new lines are represented by the hidden `\n` character and be mindful of spaces. Finally, you are provided helper functions to read and write to files, those are only here to assist and can be modified or deleted at your behest.