Assembly Parser

- Evaluates the result of your assembly code
- Calculates the total clock cycle

Tutorial

1. Add this line to the first line of the main function of your mini project 1
   - `freopen("input.txt", "w", stdout);`
   - The code above redirects your output to `input.txt`
2. Build and run your mini project 1
3. Pick a test case (ex: `1.txt`) in the directory `assembly_parser/testcase`
4. Paste the test case to the running mini project 1
5. Enter EOF to terminate the program
   - Windows: `ctrl + z`
   - Mac/Linux: `ctrl + d`
6. The program will generate the file `input.txt`
7. Build and run `main.c` in the directory `assembly_parser`
8. Paste the content of `input.txt` to the running assembly parser
9. The program will generate the file `output.txt`
10. Check `output.txt` for more information about your assembly code
    - Detail of every instruction
    - The final value of variables `x` , `y` , and `z`
    - Total clock cycles
11. You can also find the answer of every test case in `assembly_parser/testcase`
    - Note that the assembly parser uses 0 as the initial value of `x` , `y` , and `z`
    - However, `x` , `y` , and `z` will not initially be 0 in demo test cases
      - You should load their value from the memory first