COS**341** **Project 3** (2017): ***Executable Code*** *for* ***SPL***

(Part 3***c***)

Project 3: Overview

- Part 3*a*: Intermediate Code Preparation
  (done)
- Part 3*b*: Intermediate Code Generation
  (done)
- Part 3***c***: Intermediate Code **Optimisation**
  (**now**)

Given:

- **any correct and executable BASIC program,** which has been generated automatically by your intermediate code generator (in the previous part ***3b*** of this project), and a
- ZIP-folder **OPTIM.*zip*** which contains **two lessons** on code opimisation:
  - **Optim_I.*pdf***
  - **Optim_II.*pdf***

Tasks (1):

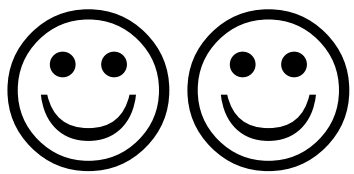
- Together with your project partner, study those two lessons in your own time, as far as they are “applicable” to BASIC as our Intermediate Code Language.
  - **Hints**:
    - Not all the stuff in those two lessons is applicable to BASIC
    - Ignore whatever is not applicable

Tasks (2):

- Implement a BASIC code optimizer with those **techniques from the two lessons** which are applicable and relevant to our COS341 Project.
  - **Input**: a NON-optimized BASIC program file B
  - **Output**: an Optimized BASIC program file B'
    - such that B' will "**run faster**" than B
- **Hints**:
  - If your Optimizer is **correctly implemented**, then both **B and B' must be semantically identical**:
    - Let **i** be an "input" and **r** a "result" from the original B,
    - Then B(**i**) → **r** ← B'(**i**) **must** be true! (same "in"→ same "out")
  - You can test this empirically in your BASIC interpreter environment for many different test-cases.

And **now**...

HAPPY PAIR-**CODING**!



Note: Plagiarism is forbidden!
Code swapping with other pairs of project students is also not allowed