Scalable and Distributed Computing
HCMC International University
Assoc. Prof. Tran Manh Ha

Course: IT139IU
Date: 2021-10-09
Time: 3 hours

# Lab #1

This session aims at learning Python programming. The focuses are on understanding Python fundamentals, multithreading and multiprocessing concepts, then writing Python programs to do some particular tasks.

**Problem 1.1:** *Running Python* (0 points)

1. Using *terminal* to run command lines:

```
$python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 3+4
7
```

2. Using *editor*, such as Notepad++, VSCode, Atom, . . . to write programs:

```
$ python hello.py
Hello, World!
```

3. Using online programming platforms such as colab.research.google.com

**Problem 1.2:** *Prime program* (5 points)

Write a Python program that accepts N positive integral numbers from the command line and verifies whether those numbers are prime. At the end, the program prints out each number with the answer. An execution of the program on the command line might look like this:

```
$ python3 prime.py 12 3 19 4
12 is not a prime
3 is a prime
19 is a prime
4 is not a prime
```

The program executes in two modes: (i) using single process/thread; (2) using more than one thread (or more than one process) for the input, for example: one thread (or process) checks numbers at even positions, and another thread (or process) checks numbers at odd positions.

**Problem 1.3:** *Symmetric program* (5 points)

Write a Python program that accepts N positive integral numbers > 9 from the command line and verifies whether those numbers are symmetric numbers. A symmetric number contains identical digits at symmetric positions, e.g., 515, 1221, 3443, etc. At the end, the program prints out each number with the answer. An execution of the program on the command line might look like this:

```
$ python3 symnum.py 111 123 4554
111 is a symmetric member
123 is a not symmetric member
4554 is a symmetric member
```

The program executes in two modes: (i) using single process/thread; (2) using N threads (or processes) responsible for N input numbers.

The programs must handle error situations (including wrong input) in a meaningful way. The solution (only one .py file) is formatted in *name_id_l1.py* and submitted to the Blackboard system before 17:30 on the same lab day. Note that students are responsible for missing/duplicated files due to wrong formats. Copying the whole source code from various sources such as the Internet is disallowed.