

## Assignment 06 - User input and while loops

1. (*execute a Python script*) Trong những bài tập trước chúng ta đã thực hành việc viết chương trình trong *Jupyter notebook* và thực thi đoạn chương trình đó bằng cách sử dụng tổ hợp phím Ctrl+Enter. Ngoài cách này, chương trình có thể được thực thi bằng cách thực thi lệnh

```
python file.py
```

trên *terminal* trong hệ điều hành Linux/macOS (hoặc trên *command prompt* trong hệ điều hành Windows).

(a) Viết một chương trình bằng ngôn ngữ lập trình Python, lưu chương trình trong một file Python (với định dạng .py) rồi thực thi chương trình qua *terminal* (hoặc *command prompt*).

(b)

```
1 for i in range(1, int(1e10)+1):
2     print(i)
3
```

Listing 1: Chương trình in các số từ 1 đến 10 tỷ

Đoạn chương trình (1) cần nhiều hơn một phút để thực thi. Làm thế nào để dừng một chương trình đang chạy trên *terminal/command prompt*?

Trình bày câu trả lời trong một file *Jupyter notebook* với tên được đặt theo mẫu

Assignment06\_FullName\_ExecuteScript.ipynb

2. (*infinite loop*)

Đoạn chương trình (2) chứa một vòng lặp vô hạn, do điều kiện trong *while* luôn đúng.

```
1 foo = 1
2 while foo > 0:
3     foo += 1
4
```

Listing 2: Một vòng lặp vô hạn trong Python

Viết một vòng lặp vô hạn với số kí tự ít nhất có thể và lưu chương trình trong một file Python với tên được đặt theo mẫu Assignment06\_FullName\_InfiniteLoop.py

3. (*rock-paper-scissors game*)

(a) Viết một chương trình để hai người có thể chơi trò chơi kéo - búa - bao (oẳn tù tì). Lưu chương trình trong một file Python với tên được đặt theo mẫu

Assignment06\_FullName\_RockPaperScissors.py

(b) Cách tiếp cận đơn giản là cho phép hai người chơi *A, B lần lượt* đưa ra lựa chọn và nhập nó vào chương trình thông qua bàn phím (giả sử *A* nhập trước). Cách chơi như vậy không công bằng vì ở mỗi lượt chơi *B* luôn biết được lựa chọn của *A trước khi B* đưa ra lựa chọn của mình. Có thể hiểu tính công bằng theo nghĩa nào khác? Làm thế nào để đảm bảo tính công bằng cho trò chơi?

*Fun fact*: “Oẳn tù tì” có phát âm gần giống với “one two three”.

4. (*sign up an account*)

Viết một chương trình cho phép người dùng tạo một tài khoản với tên đăng nhập (username) và mật khẩu (password) và yêu cầu người dùng xác nhận lại mật khẩu đã nhập. Lưu chương trình trong một file Python với tên được đặt theo mẫu Assignment06\_FullName\_SignUpAccount.py

## Additional Problems

### 5. (largest prime factor)

Giải quyết bài toán trong **Project Euler problem 3**. Trình bày câu trả lời trong một file *Jupyter notebook* với tên được đặt theo mẫu `Assignment06_FullName_LargestPrimeFactor.ipynb`

### 6. (do-while conversion)

Vòng lặp do-while trong ngôn ngữ lập trình C++ có dạng

```
1 do {  
2 // codes;  
3 }  
4 while (testExpression);  
5
```

Listing 3: Cấu trúc của vòng lặp do-while trong C++

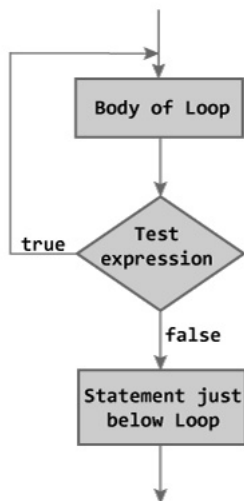


Figure: Flowchart of do...while Loop

Hình 1 là sơ đồ thực thi của loại vòng lặp này. Chương trình thực thi khối lệnh trong phần do, sau đó kiểm tra điều kiện ở phần while. Nếu điều kiện xảy ra thì chương trình quay trở lại điểm xuất phát vòng lặp, nếu điều kiện không xảy ra thì chương trình thoát khỏi vòng lặp.

- (a) So sánh vòng lặp do-while trong C++ với vòng lặp while trong Python.
- (b) Viết lại đoạn chương trình (3) ở bài tập 6 bằng ngôn ngữ lập trình Python.

Trình bày câu trả lời trong một file *Jupyter notebook* với tên được đặt theo mẫu `Assignment06_FullName_DoWhileConversion.ipynb`

Hình 1: [Programiz.com](https://www.programiz.com)

### 7. (unstoppable program)

Trong bài tập 1, ta đã tìm hiểu những cách để dừng một chương trình đang được thực thi trên *terminal/command prompt*.

Một cách khả thi là dừng chương trình hiện tại bằng tổ hợp phím `Ctrl + C`. Một cách khác là đóng cửa sổ *terminal/command prompt*. Tuy nhiên việc đóng cửa sổ không được khuyến khích vì làm mất những thông tin hiện có của cửa sổ đó.

Hãy viết một chương trình sao cho một khi chương trình đó bắt đầu được thực thi, cách duy nhất để dừng nó là đóng cửa sổ hiện tại. Lưu chương trình trong một file Python với tên được đặt theo mẫu `Assignment06_FullName_UnstoppableProgram.py`

Gợi ý: sử dụng vòng lặp vô hạn và câu trúc `try/except`.

### 8. (banking simulation)

Nối tiếp bài tập 4, trong bài tập này ta sẽ xây dựng một chương trình mô phỏng hoạt động của một ngân hàng. Đầu tiên, cải tiến việc tạo tài khoản bằng cách thực hiện các yêu cầu sau:

- (a) Cung cấp cho người dùng những hướng dẫn phù hợp trong quá trình người dùng tương tác với chương trình.
- (b) Nêu những tiêu chí thường được sử dụng để xác định một mật khẩu là hợp lệ. Sử dụng những tiêu chí đó để kiểm tra tính hợp lệ của mật khẩu.
- (c) Nhằm đảm bảo tính an toàn của tài khoản, yêu cầu người dùng nhập lại mật khẩu nếu nó không hợp lệ.
- (d) Ẩn đi mật khẩu trong quá trình người dùng nhập mật khẩu.

Ngoài việc tạo tài khoản với tên đăng nhập và mật khẩu, mô phỏng này cần:

- (a) Trong quá trình tạo tài khoản, cho phép người dùng cung cấp thêm các thông tin bổ sung.
- (b) Lưu trữ thông tin của tất cả người dùng (ngay cả khi chương trình đã dừng).
- (c) Cho phép người dùng sửa đổi thông tin của chính họ.
- (d) Ghi nhận số tiền người dùng có trong tài khoản.
- (e) Cho phép chuyển tiền giữa các tài khoản.
- (f) Gửi/rút tiền tiết kiệm.
- (g) Tạo mật khẩu mới cho một tài khoản khi được yêu cầu.
- (h) Bảo đảm bí mật những thông tin của người dùng.
- (i) ...

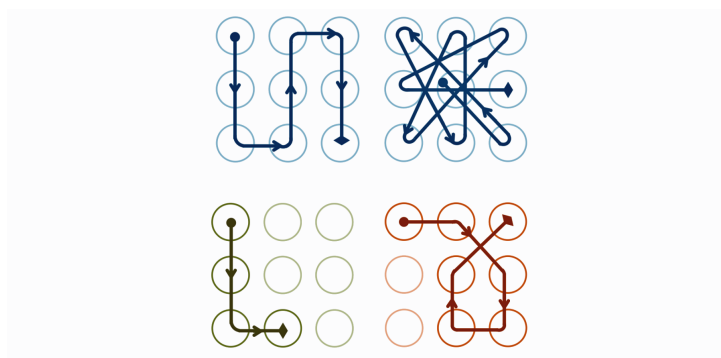
Trình bày câu trả lời một cách thích hợp và đặt tên theo mẫu

Assignment06\_FullName\_BankingSimulation

Chú ý: trong thực tế việc mở tài khoản tại ngân hàng yêu cầu người dùng cung cấp các thông tin định danh khác (ví dụ như CMND). Tuy nhiên những yếu tố đó không được tính đến ở đây.

#### 9. (4-to-9 points password)

Hãy viết một chương trình mô phỏng việc tạo mật khẩu đường gấp khúc trên điện thoại thông minh.



Hình 2: Một số mật khẩu đường gấp khúc

## Related Topics

- parsing in `argparse` module in Python.
- extracting arguments via `sys.argv` in Python.

## Inspiration

The structure of some problems in this assignment (and ones given in the future) is inspired by the concept of *Minimum Viable Product (MVP)*. By keeping the concept in mind, programs are first created to handle the most simple case, then they are gradually upgraded to deal with more sophisticated cases. Therefore, users always have a viable product to experience with and perhaps they can give developers some useful feedback on the current product.

Have a look at the articles from [Medium](#) and [Crisp Blog](#) for further understanding of *MVP*.