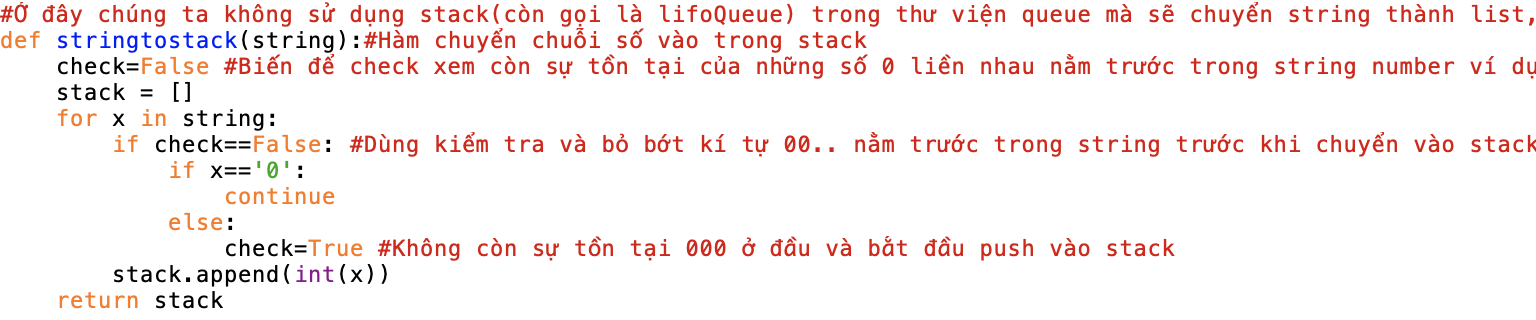
**Bài góp ý**

Chào các bạn ngày hôm nay mình muốn chia sẻ cũng như góp ý một vài điều mà mình đã học được khi mình đi thực tập. Nhân tiện đây thì mình cũng giới thiệu qua về bản thân mình, mình là sinh viên năm 4 đang học chuyên ngành khoa học máy tính tại Bách Khoa-HCM. Như các bạn biết trường đại học là nơi dạy cho chúng ta biết rất nhiều rất nhiều kiến thức, tiếc rằng kinh nghiệm làm việc không phải là thứ có thể dạy được. Hai chữ kinh nghiệm nghe có rất cơ bản, nhưng thực chất nó là thứ rất cốt lõi là những điều mà chúng ta rút ra được thông qua quá trình học tập, tiếp thu và làm việc. Kinh nghiệm có rất nhiều loại và thứ mà mình muốn chia sẻ đến các bạn đó là kinh nghiệm làm thế nào để có thể code đúng cách, chuyên nghiệp code ra một sản phẩm tốt và tối ưu nhất có thể, đảm bảo sự hài lòng cho khách hàng. Để hiểu rõ hơn thì mình xin lấy ví dụ về một đề bài mà mình đã từng được giao thực hiện.

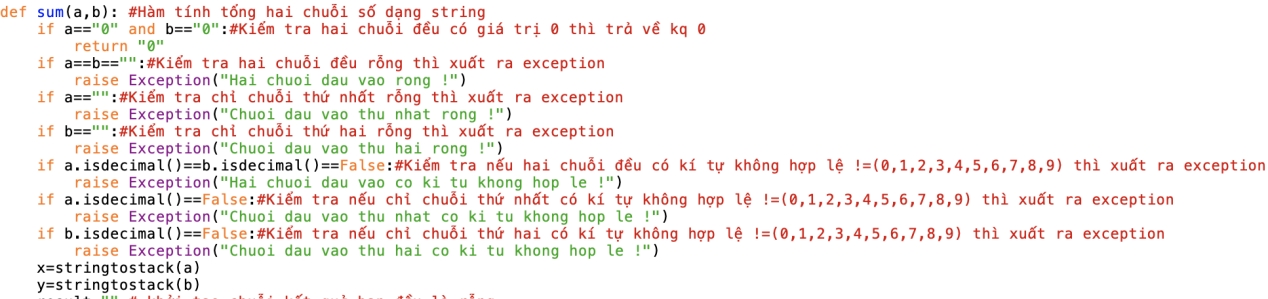
Đề bài có yêu cầu là “Bạn hãy hiện thực một hàm **Sum** (Hàm tính tổng) với đầu vào là hai chuỗi (string) số, mỗi kí tự ở trong hai chuỗi chỉ chứa kí tự 0->9 và kết quả trả ra của hàm **Sum** này là một chuỗi tổng của hai số nguyên được biểu diễn dưới dạng chuỗi, mà ta đã truyền vào trước đó” Nhìn vào đề bài này thì bạn sẽ nghĩ rằng đề bài này sao mà dễ thế không cần suy nghĩ nhiều về thuật toán để giải bài này vì ngay một đứa học sinh lớp 3 cũng biết cách làm sao để giải từng bước việc cộng hai số nguyên lại với nhau. “Chỉ cần đăt hai số lên rồi cộng lần lượt từng số một từ phải sang trái, tổng nào lớn hơn hoặc bằng 10 thì ta chỉ cần viết hàng đơn vị xuống sau đó nhớ 1 rồi cộng tiếp vào tổng của hai số tiếp theo còn nếu tổng nhỏ hơn 10 thì ta viết tổng xuống thôi, lặp lại lần lượt như vậy cuối cùng ta có tổng hai số cần tính”.

Chưa đề cập đến sai sót, hai điều quan trọng đầu tiên ở đây mình muốn nhắn nhủ là không chỉ là code cho ra đúng kết quả và gửi cho khách hàng là xong, mà đầu tiên chúng ta nên có một bài trình bày lại đúng yêu cầu chúng ta nhận từ khách hàng điều đó có thể giúp khách hàng tin tưởng rằng chúng ta nghiêm túc trong quá trình thực hiện. Và biết được cách giới thiệu cho khách hàng biết về điểm tốt mà chúng ta đã làm được trong mã nguồn của chúng (ví dụ như mình đã kiểm tra chuỗi đầu vào và cách xử lý nếu nó không hợp lệ, chương trình của chúng ta có thể tính toán với các chuỗi số rất lớn lên đến cả hàng ngàn kí tự,...)

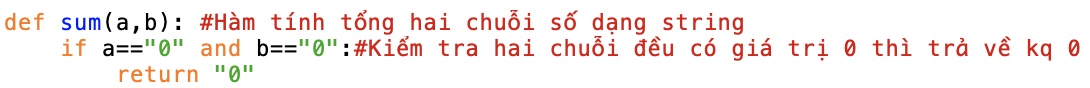
Thứ hai là những vấn đề còn chưa tốt cũng như cách khắc phục trong bài code mà mình đã hiện thực và mình muốn chia sẻ đến các bạn khi bản thân đã được chỉ bảo và nhận ra những khuyểt điểm đó.



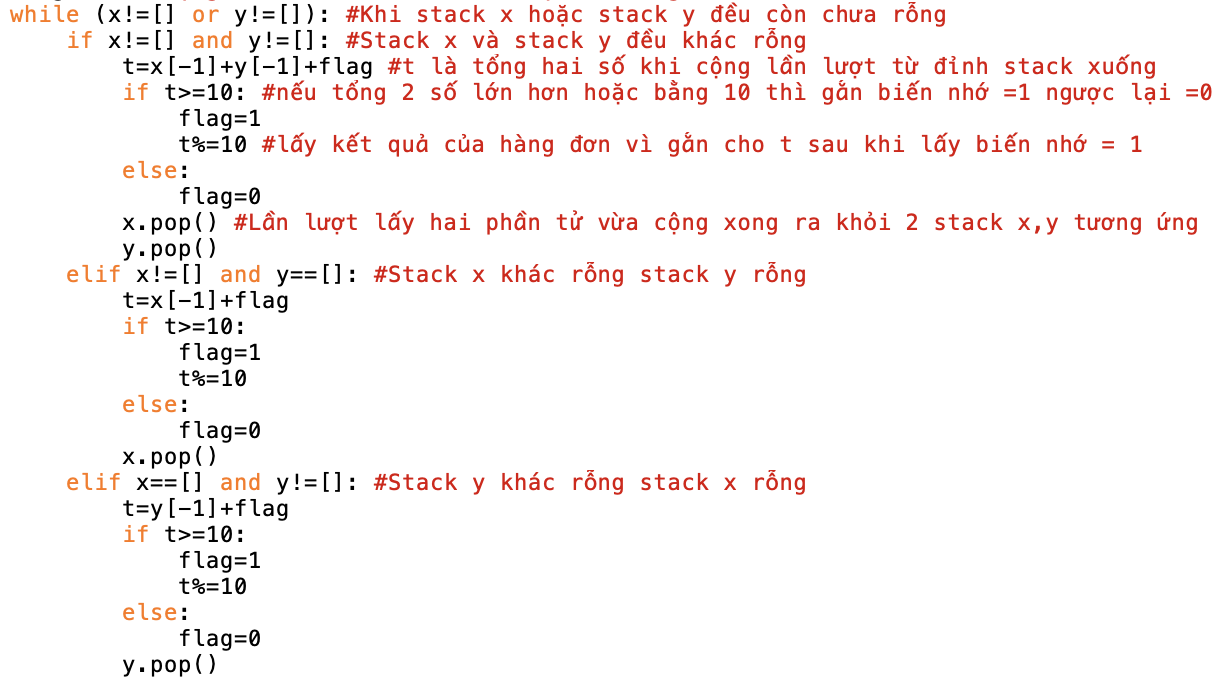
- Vấn đề thứ 1: Ở đây khi hiện thực hàm sum thay thì mình tính toán trực tiếp trên 2 chuỗi truyền vào thì mình đã chuyển chúng thành 2 stack sau đó mới thực hiện tính toán rồi lại chuyển kết quả về dạng chuỗi, điều đó không chỉ làm cho chương trình mình trở nên phức tạp hơn mà còn làm thời gian chạy của nó tăng lên. Và nguyên nhân là mình không có thiết kế một sơ đồ logic (flow char) đúng cho việc xử lý bài toán trên trước khi bắt tay vào hiện thực mã nguồn.



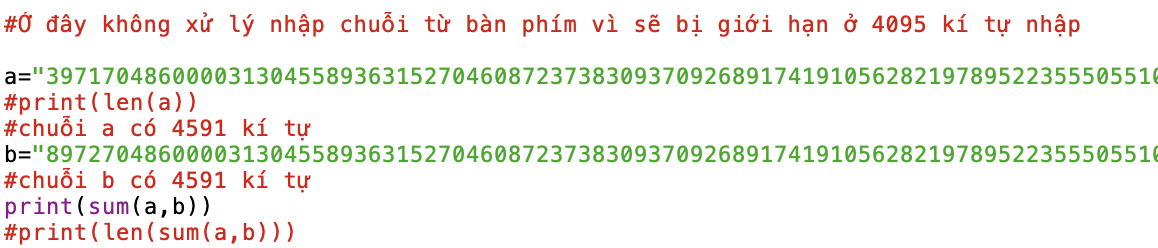
- Vấn đề thứ 2: Mình đã đưa phần xử lý những điều kiện lỗi của hai chuỗi đầu vào vào trong phần xử lý logic chính của chương trình (ở đây chính là hàm **Sum**), đáng lẽ ra chúng ta phải đưa phần xử lý lỗi vào trong một phần riêng (có thể là viết vào trong một hàm riêng chẳng hạn và hàm đó chỉ có chức năng duy nhất là kiểm tra lỗi đầu vào mà thôi) tuyệt đối không để chung với phần xử lý logic của bài toán.



- Vấn đề thứ 3: Trong phần kiểm tra đầu vào mình đã thực hiện phần việc kiểm tra (Hai chuỗi đầu vào có giá trị bằng 0 và trả về kết quả 0) mà đáng lẽ điều này nên được thực thi trong phần logic tính toán chính của chương trinh, điều này đã khiến cho logic của cả chương trình mình viết bị sai và nó có thể khiến cho thời gian chạy chương trình của mình tăng gấp hai lần.



- Vấn đề thứ 4: Ở câu lệnh điều kiện if-else mặc dù cùng một logic xử lý là: “Lấy ra kết quả khi tính lần lượt để viết xuống và giá trị nhớ rồi sau đó bỏ hai phần tử đó ra khỏi stack của chúng” nhưng mình lại chia ra xử lý nó đến 3 lần trong một vòng lặp. Nó dẫn đến phần code xử lý logic chính của chúng ta bị dư thừa ra.



- Vấn đề thứ 5: Mình không có bằng chứng về việc mình đã test kĩ chức năng của hàm **Sum** sau khi mình hiện thực xong nó, đúng là như vậy nhìn vào đoạn code của mình ở trên thì bạn có thể thấy mình có viết thử một một cặp chuỗi số để test mà thôi mặc dù mình có bảo là mình đã xóa đi rồi test lại với những cặp số khác nhau hàng trăm nghình lần thì ai tin chứ, vì mình không có đưa ra được bằng chứng chứng mình việc đó. Và nguyên nhân chính ở đây chính là do mình chưa biết áp dụng phương pháp unit-test nên khiến cho chất lượng chương trình của mình không được đảm bảo và chắc chắn.

=> Đúc kết lại khi chúng ta nhận làm một dự án dù lớn hay nhỏ đến cỡ nào thì chúng ta cũng cần phải nắm rõ, nhớ kĩ các yêu cầu của khách hàng, biết được những điểm tốt trong mã nguồn của chúng , biết cách khắc phục những điểm yếu và hạn chế trong quy trình thực hiện cũng như với mã nguồn của chúng ta, hãy học và biết thiết kế các sơ đồ logic cho bài toán ( như flow char, activity-diagram, sequence-diagram, class-diagram,...) trước khi bắt tay vào hiện thực mã nguồn, phần xử lý logic cho chức năng chính của bài toán cũng như những chức năng phụ nên được viết tách ra thành các phần riêng, sủa lại cách viết để logic trong mã nguồn không bị thừa, cần có kĩ năng về testing(testcoverage)

Mình hi vọng sau bài góp ý này của mình, nó có thể giúp đỡ cho các bạn trong việc cải thiện phần nào các kỹ năng của bản thân mình. Và chính mình cũng đang trên con đường cố gắng khắc phục những nhược điểm nêu trên để có thể ngày một tốt hơn. Nếu các bạn có những suy nghĩ, đóng góp về những điều thật bổ ích hơn nữa thì mình hi vọng các bạn hãy dành chút thời gian quý giá của mình để có thể chia sẻ chúng đến với mọi người xung quanh.