**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

A picture containing icon

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**Chương trình làm bài trắc nghiệm**

**PHẠM VĂN THÁI – 20206303**

**Mã lớp: 133590**

thai.pv20206303@sis.hust.edu.vn

**Ngành Hệ thống thông tin quản lý**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Vũ Thành Nam  Chữ ký của GVHD |
| **Bộ môn:** | Kỹ thuật lập trình |
| **Viện:** | Toán Ứng dụng và Tin học |
| **HÀ NỘI, 8/2022** | |

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

**Lời cảm ơn**

Kỹ thuật lập trình là môn học bắt buộc đối với chuyên ngành em đang theo học – ngành Hệ thống thông tin quản lý. Theo cảm nhận cá nhân, em cảm thấy, môn học này khá trừu tượng và khó hiểu nếu tự học đặc biệt là đối với những sinh viên chưa có nhiều kinh nghiệm trong các dự án và các kỹ năng lập trình. Nhưng nhờ có sự hướng dẫn, giảng dạy của thầy trên giảng đường và thông qua các trao đổi trên online hay offline, thầy đã giúp chúng em làm rõ các nội dung cũng như nắm bắt được tinh thần của môn học. Những kiến thức thầy truyền tải không dừng lại ở lý thuyết mà còn nhiều phần mở rộng, phân tích, trao đổi và phản biện. Em cảm ơn thầy trong suốt 16 tuần học vừa qua, cũng như sự giải đáp thắc mắc của thầy để em cùng các bạn có thể làm tốt phần báo cáo này.

**Tóm tắt nội dung báo cáo**

Đề tài bài tập lớn của em là Xây dựng chương trình chương trình làm bài trắc nghiệm. Hiện nay, các trường học như các trường cấp trung học cơ sở, trường trung học phổ thông cũng như các trường đại học đã và đang tích hợp thêm hình thức thi trắc nghiệm trực tuyến. Việc thi trực tuyến như vậy có rất nhiều lợi thế cho cả thí sinh và giáo viên. Kết quả bài thi đảm bảo tính chính xác, thời gian chấm thi được rút ngắn và không bị giới hạn về mặt không gian (điều này thể hiện rõ nhất trong đại dịch Covid-19). Nhận thấy, đây là đề tài có tính ứng dụng cao và tập trung vào quản lý và nghiệp vụ, để áp dụng trong thực tế cần nhiều yếu tố và rất phức tạp, vì thế, chương trình cá nhân này chỉ thể hiện được cấu trúc, các tính năng cơ bản của một chương trình làm bài trắc nghiệm và cố gắng vận dụng các kỹ thuật lập trình được học. Chương trình bao gồm: đọc dữ liệu (câu hỏi và đáp án) từ file text, sinh đề và hiển thị câu hỏi, nhập phương án trả lời (giả định chọn một trong nhiều đáp án), hiển thị kết quả và có thể lưu bài làm, kết quả. Chương trình được xây dựng theo hướng lập trình hướng đối tượng. Mã nguồn được viết bằng ngôn ngữ lâp trình Python, được chạy trên Command Promp và Text Editor “Sublime Text”. Sau khi hoàn thành cùng với các kết quả kiểm thử, chương trình đã đáp ứng được các yêu cầu thực tế đặt ra ở mức độ đáp ứng đủ và cơ bản. Để áp dụng được trong thực tế, cần tích hợp và nâng cấp thêm các chức năng, và tùy theo yêu cầu của nghiệp vụ. Ví dụ như chương trình LMS có thể thêm tính năng làm bài trắc nghiệm tối đa 3 lần, lưu đáp án vào 3 file và ghi nhận điểm cao nhất là điểm của thí sinh, tích hợp thêm thời gian thực,… Các kỹ năng lập trình được sử dụng như module hóa chương trình, xây dựng các hàm tốt, viết code hiệu quả, dễ hiểu, rõ ràng, comment đúng và đủ,…

Sinh viên thực hiện

Ký và ghi rõ họ tên

**MỤC LỤC**

[CHƯƠNG 1. CẤU TRÚC CHƯƠNG TRÌNH 1](#_Toc110260238)

[1.1 Giới thiệu chung 1](#_Toc110260239)

[1.2 File cau\_hoi.txt 1](#_Toc110260240)

[1.3 File luu\_ket\_qua.txt 1](#_Toc110260241)

[1.4 File trac\_nghiem.py 1](#_Toc110260242)

[1.4.1 Mô tả Class 2](#_Toc110260243)

[1.4.2 Mô tả các hàm 2](#_Toc110260244)

[1.4.3 Mô tả chương trình chính 5](#_Toc110260245)

[CHƯƠNG 2. CÁC KỸ THUẬT LẬP TRÌNH 7](#_Toc110260246)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN CHƯƠNG TRÌNH 11](#_Toc110260247)

[TÀI LIỆU THAM KHẢO 21](#_Toc110260248)

# CẤU TRÚC CHƯƠNG TRÌNH

## Giới thiệu chung

Đối với chương trình “Thi trắc nghiệm”, em sử dụng ngôn ngữ lập trình Python.

Chương trình trắc nghiệm sẽ gồm 3 file:

File trac\_nghiem.py: chứa mã nguồn, hàm main để thực thi.

File “cau\_hoi.txt”: có nhiệm vụ chứa phần câu hỏi và đáp án để sinh đề ngẫu nhiên cho thí sinh.

File “luu\_ket\_qua.txt”: Sau khi thí sinh làm bài xong, kết quả của bài thi sẽ được lưu tại đây.

Ngoài ra, còn có 3 file, format tương tự file “cau\_hoi.txt”, được thay đổi đổi đôi chút để phục vụ mục đích kiểm thử.

File “cau\_hoi\_sai\_1.txt”.

File “cau\_hoi\_sai\_2.txt”.

File “cau\_hoi\_7.txt”: file chứa 7 câu hỏi trắc nghiệm.

## File cau\_hoi.txt

Chức năng: Lưu dữ liệu câu hỏi, các đáp án của câu hỏi và đáp án đúng của câu hỏi để thực hiện sinh đề ngẫu nhiên.

Các tổ chức file:

Dòng thứ nhất: Số câu.

Mỗi 6 dòng tiếp theo lưu dữ liệu của một câu. Trong 6 dòng đó:

Dòng 1: Lưu dữ liệu câu hỏi.

4 dòng tiếp theo: Lưu các đáp án của câu hỏi.

Dòng 6: Lưu đáp án đúng của câu hỏi.

## File luu\_ket\_qua.txt

Chức năng: Lưu dữ liệu bài làm của thí sinh, bao gồm họ và tên, mã số sinh viên, lớp, đáp án sinh viên và đáp án đúng của câu hỏi(căn cứ vào đó để có thể thực hiện phúc khảo sau này) và điểm của sinh viên.

Cách tổ chức file:

Mỗi 6 dòng (tính từ dòng 1) lưu dữ liệu bài làm của một thí sinh, bao gồm:

Dòng 1: Lưu dữ liệu họ và tên thí sinh.

Dòng 2: Lưu dữ liệu mã số sinh viên.

Dòng 3: Lưu dữ liệu lớp thí sinh.

Dòng 4: Đáp án sinh viên đã chọn (Nếu thí sinh bỏ trống đáp án thì đáp án được lưu là “-”).

Dòng 5: Đáp án đúng.

Dòng 6: Điểm thí sinh.

## File trac\_nghiem.py

Chương trình xây dựng theo hướng lập trình hướng đối tượng.

### Mô tả Class

***Chương trình bao gồm: 2 Lớp đối tượng (class)***

Class **SinhVien**:

Bao gồm 6 thuộc tính: ho\_ten, mssv, lop, tc\_dap\_an, tc\_dap\_an\_dung, diem. Các thuộc tính này là những thông tin cơ bản cần có cho một bài thi trắc nghiệm.

Sau khi một thí sinh thực hiện xong bài thi, chương trình tạo ra một đối tượng của sinh viên mà dựa vào đó, ta có dữ liệu cần thiết để hiện thị kết quả và lưu thông tin bài làm.

Class **Cau**:

Bao gồm 3 thuộc tính: cau\_hoi, tat\_ca\_dap\_an, dap\_an\_dung.

Sau khi đọc dữ liệu từ file “cau\_hoi.txt”, chương trình sẽ tạo ra một mảng các đối tượng thuộc lớp “Cau”. Từ đó, chương trình sẽ sinh đề ngẫu nhiên (bao gồm cả sinh đáp án đúng) và chấm bài thi theo đáp án đó.

### Mô tả các hàm

***Chương trình bao gồm 15 hàm (function):***

**doc\_file**:

Return:

tat\_ca\_cau: mảng các câu hỏi trắc nghiệm( phần tử của mảng là đối tượng thuộc lớp ***Cau*).**

Mô tả: Chương trình tiến hành đọc file ***“cau\_hoi.txt”***.

**kiem\_tra\_cau\_hoi**:

Tham số đầu vào:

tat\_ca\_cau: mảng các câu hỏi trắc nghiệm( phần tử của mảng là đối tượng thuộc lớp ***Cau*).**

Return: True hoặc False.

Mô tả: Kiểm tra từng phần tử trong mảng câu hỏi trắc nghiệm, nếu phần tử nào có thuộc tính bằng None thì trả về False. Nếu các thuộc tính đều khác None trả về True.

**xao\_tron\_cau**:

Tham số đầu vào:

tat\_ca\_cau: mảng các câu hỏi trắc nghiệm( phần tử của mảng là đối tượng thuộc lớp ***Cau*).**

Return: mảng các đáp án đúng của bộ đề được sinh ngẫu nhiên.

Hàm cũng sẽ in đề sau khi được sinh ngẫu nhiên.

Mô tả: Sau khi có được mảng các câu hỏi trắc nghiệm đúng theo yêu cầu, tiến hành xáo trộn các phần tử trong mảng và xáo trộn mảng tất cả đáp án trong từng phần tử đó.

**nhap\_thong\_tin\_ca\_nhan**:

Return:

ho\_ten: họ và tên thí sinh.

mssv: mã số sinh viên thí sinh.

lop: lớp của thí sinh.

Mô tả: Hàm yêu cầu nhập họ và tên, mã số sinh viên, lớp của thí sinh.

**in\_thong\_tin\_ca\_nhan(sinh\_vien)**:

Tham số đầu vào:

sinh\_vien: đối tượng của lớp ***SinhVien***.

Hàm (thủ tục) in ra thông tin thí sinh bao gồm: Họ và tên, mã số sinh viên và lớp.

**sua\_thong\_tin\_ca\_nhan(sinh\_vien)**:

Tham số đầu vào:

sinh\_vien: đối tượng của lớp ***SinhVien***.

Return: sinh\_vien

Mô tả: Hàm cho lựa chọn các thay đổi trong 3 thuộc tính của ***SinhVien*** bao gồm: ***ho\_ten***, ***mssv***, ***lop***.

**lua\_chon\_so(promp, min\_, max\_)**:

Tham số đầu vào:

promp: Chuỗi kí tự để hỏi.

min\_: số nhỏ.

max\_: số lớn.

Return: số nguyên thỏa mãn trong khoảng từ số nhỏ đến số lớn.

Mô tả: Hàm yêu cầu nhập số nguyên trong khoảng cho trước. Mọi số không thỏa mãn hay các kí tự, chuỗi kí tự được nhập đều được yêu cầu nhập lại cho đến khi nhập đúng.

**dap\_an\_hop\_le(promp, array)**:

Tham số đầu vào:

promp: Chuỗi kí tự để hỏi.

array: mảng các lựa chọn.

Return: lựa chọn hợp lệ.

Mô tả: Hàm yêu cấu nhập lựa chọn theo gợi ý từ ***promp***. Mọi lựa chọn không thuộc trong ***array*** đều được yêu cầu nhập lại cho đến khi nhập đúng.

**nhap\_dap\_an(tat\_ca\_cau)**:

Tham số đầu vào:

tat\_ca\_cau: mảng các câu hỏi trắc nghiệm( phần tử của mảng là đối tượng thuộc lớp ***Cau*).**

Return: mảng tất cả đáp án thí sinh đã nhập.

Mô tả: Hàm yêu cầu thí sinh nhập đáp án từ câu 1 cho đến câu cuối cùng. Hàm chỉ chấp nhập các đáp án bao gồm: “A”, “B”, “C”, “D” và “”(None) bằng việc sử dụng hàm ***dap\_an\_hop\_le***(đã nêu ở trên).

**hien\_dap\_an(tc\_dap\_an)**:

Tham số đầu vào:

tc\_dap\_an: tất cả đáp án của thí sinh nhập.

Mô tả: Các đáp án của thí sinh được lưu vào một mảng và thực hiện vòng for để in giá trị từng phần tử trong mảng đó. In đáp án thí sinh đã nhập.

**sua\_dap\_an(tc\_dap\_an)**:

Tham số đầu vào:

tc\_dap\_an: tất cả đáp án của thí sinh nhập.

Return: mảng tất cả đáp án của thí sinh sau khi sửa.

Mô tả:

Lựa chọn câu muốn sửa bằng hàm ***lua\_chon\_so***(đã nêu ở trên).

Lựa chọn đáp án thay thế cho câu muốn sửa bằng hàm ***dap\_an\_hop\_le***(đã nêu ở trên).

**luu\_dap\_an(sinh\_vien):**

Tham số đầu vào:

sinh\_vien: đối tượng của lớp ***SinhVien***.

Mô tả: Sau khi thí sinh làm xong bài kiểm tra, tạo ra một đối tượng thuộc lớp SinhVien với đầy đủ thông tin thuộc tính, hàm sẽ viết các thuộc tính của thí sinh ra file ***luu\_ket\_qua*** bao gồm ho\_ten, mssv, lop, diem, tc\_dap\_an, tc\_dap\_an\_dung.

**tinh\_diem(tc\_dap\_an, tc\_dap\_an\_dung):**

Tham số đầu vào:

tc\_dap\_an: mảng chứa đáp án thí sinh đã điền.

tc\_dap\_an\_dung: mảng chứa đáp án đúng của đề trắc nghiệm sinh ngẫu nhiên.

Return: điểm của bài thi.

Mô tả: Hàm sẽ so sánh từng phần tử của mảng ***tc\_dap\_an*** so với từng phần tử của mảng ***tc\_dap\_an\_dung*** theo thứ tự. Nếu trùng nhau, điểm được cộng ngược lại thì điểm giữ nguyên.

**hien\_thi\_ket\_qua(sinh\_vien):**

Tham số đầu vào:

sinh\_vien: đối tượng của lớp ***SinhVien***.

Mô tả: Sau khi thực hiện hàm ***luu\_ket\_qua*** (đã nêu ở trên), hàm sẽ in ra các thông tin về bài làm của thí sinh bao gồm: Họ và tên sinh viên, mã số sinh viên và lớp sinh viên.

**show\_menu():**

Mô tả: Hiển thị ra màn hình các tính năng mà sinh viên có thể lựa chọn để làm bài bao gồm: Hiển thị bài làm, Sửa đáp án, Sửa thông tin sinh viên và Lưu đáp án (Chỉ lưu được một lần).

### Mô tả chương trình chính

Các hàm đã được trình bày chi tiết phía trên, các bước trong hàm main có sử dụng các hàm xây dựng sẽ lược bớt phần tham số đầu vào.

Các hàm trong chương trình đã được giải thích chi tiết phía trên.

Các biến được sử dụng trong hàm main(): ***tat\_ca\_cau, kiem\_tra, ho\_ten, mssv, lop, diem, sinh\_vien, choice, diem, lua\_chon, tat\_ca\_dap\_an\_dung, tat\_ca\_dap\_an***.

Bước 0: Khởi tạo ***kiem\_tra*** = None.

Bước 1: Kiểm tra lỗi đọc file “cau\_hoi.txt” bằng hàm **try**, **except.**

Thực hiện hàm **doc\_file.**

kiem tra = kiem\_tra\_cau\_hoi(tat\_ca\_cau).

Nếu xảy ra lỗi đọc file, in ra "Kiem tra lai file cau\_hoi.txt”.

Bước 2: Nếu kiem\_tra = True, tức bộ đề đạt yêu cầu (đầy đủ câu hỏi, câu hỏi đầy đủ 4 đáp án A, B, C, D), thực hiện khối lệnh sau:

Bước 2.1:

Thực hiện sinh đề bằng cách thực hiện ***tat\_ca\_dap\_an\_dung*** = **xao\_tron\_cau**().

Nhập thông tin cá nhân của thí sinh, đáp án của thí sinh khi làm bài, do đó tạo ra đối tượng ***sinh\_vien*** với đầy đủ thông tin.

Thực hiện vòng lặp vô hạn, để chọn các tính năng làm bài (*gồm 4 tính năng: Hiển thị đáp án, Sửa đáp án, Sửa thông tin cá nhân, Lưu đáp án*), nhằm thay đổi thuộc tính của ***sinh\_vien***.

Bước 2.1.0: Chọn tính năng để làm bài ***choice*** = **lua\_chon\_so**().

Bước 2.1.1: Nếu ***choice*** = 1.

Hiển thị thông tin các nhân và đáp án đã điền của thí sinh: **in\_thong\_tin\_ca\_nhan**(), **hien\_dap\_an**().

Bước 2.1.2: Nếu choice = 2:

Thực hiện sửa đáp án: **sua\_dap\_an**().

Cập nhật lại điểm theo đáp án đã sửa: ***diem***= **tinh\_diem**().

Cập nhật lại đối tượng ***sinh\_vien*** và hiển thị đáp án đã sửa.

Bước 2.1.3: Nếu choice = 3:

Thực hiện sửa thông tin thí sinh: **sua\_thong\_tin\_ca\_nhan**().

Bước 2.1.4: Nếu choice = 4:

Lưu đáp án vào file “luu\_ket\_qua”: **luu\_dap\_an**().

Hiển thị kết quả bài làm của sinh viên: **hien\_thi\_ket\_qua**()**.**

Bước 3: Nếu ***kiem\_tra*** = False, in ra màn hình "Kiem tra lai file cau\_hoi.txt da day du thong tin de bai chua”.

# CÁC KỸ THUẬT LẬP TRÌNH

Đặt tên file, tên các biến, tên Class rõ ràng, có tính gợi nhớ, gợi tả.

Chương trình cần 3 file: 1 file chương trình chính và 2 file txt, tên file được đặt rõ ràng, thể hiện được chức năng của file đó.

Tên các hàm, tên biến, tên Class được đặt theo quy định chung:

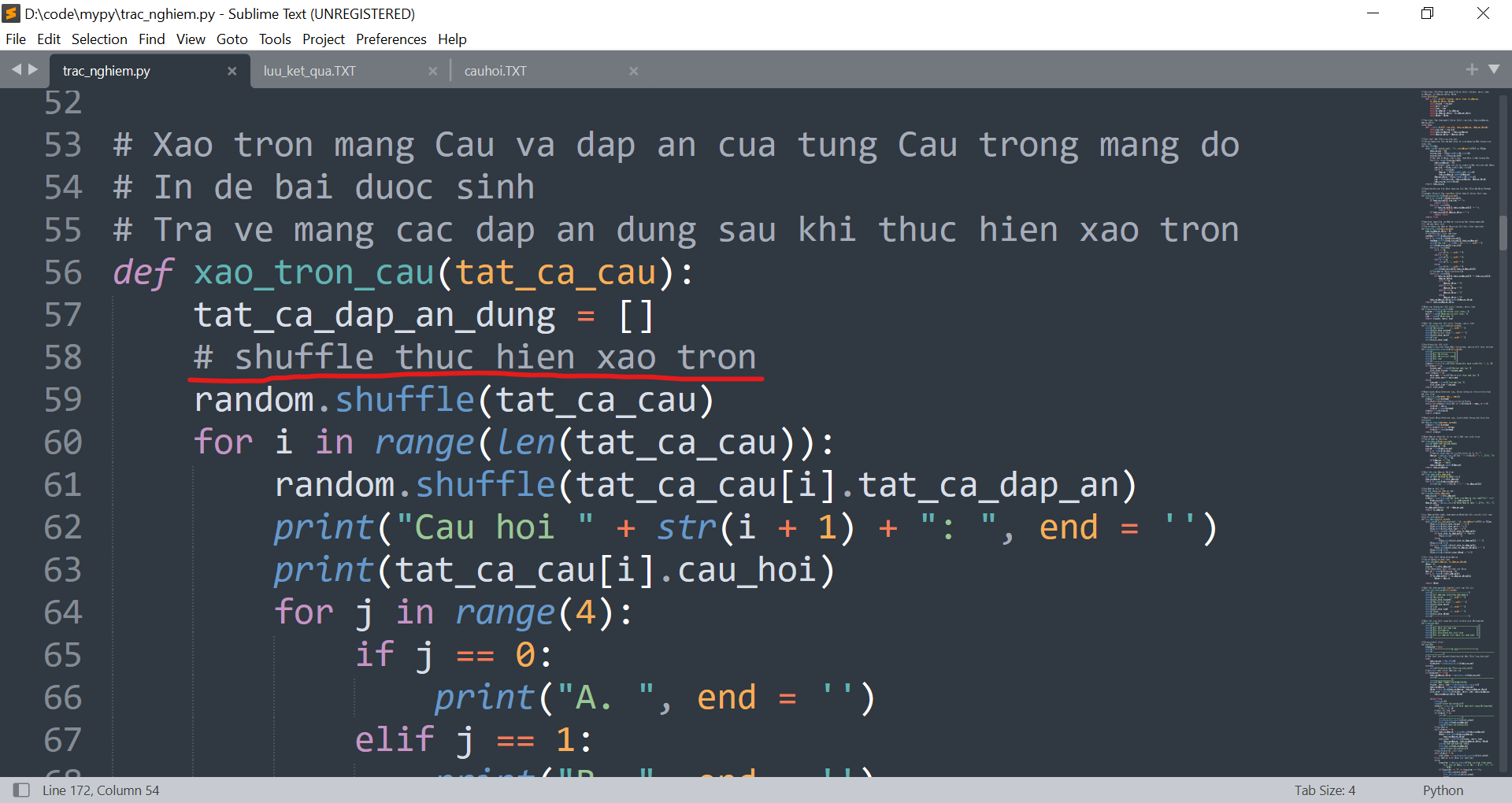
Tên hàm, tên biến: được viết bằng chữ thường, cách nhau bởi dấu \_, tên hàm thường là động từ để thể hiện chức năng.

Tên Class: Viết hoa chữ cái đầu của mỗi từ.

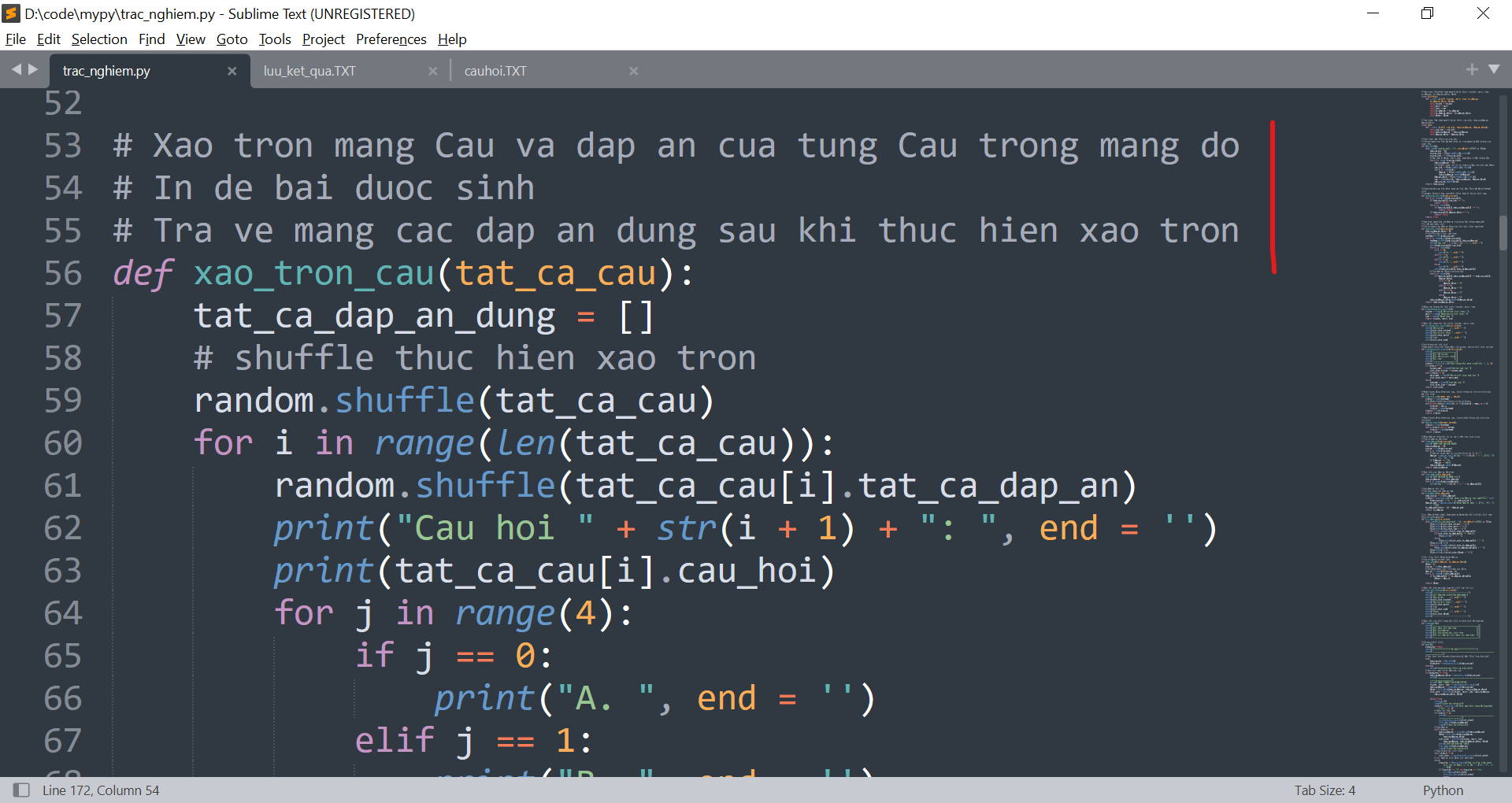
Chương trình theo hướng lập trình hướng đối tượng, chương trình chính thao tác, xoay quanh trên đối tượng SinhVien.

Comment:

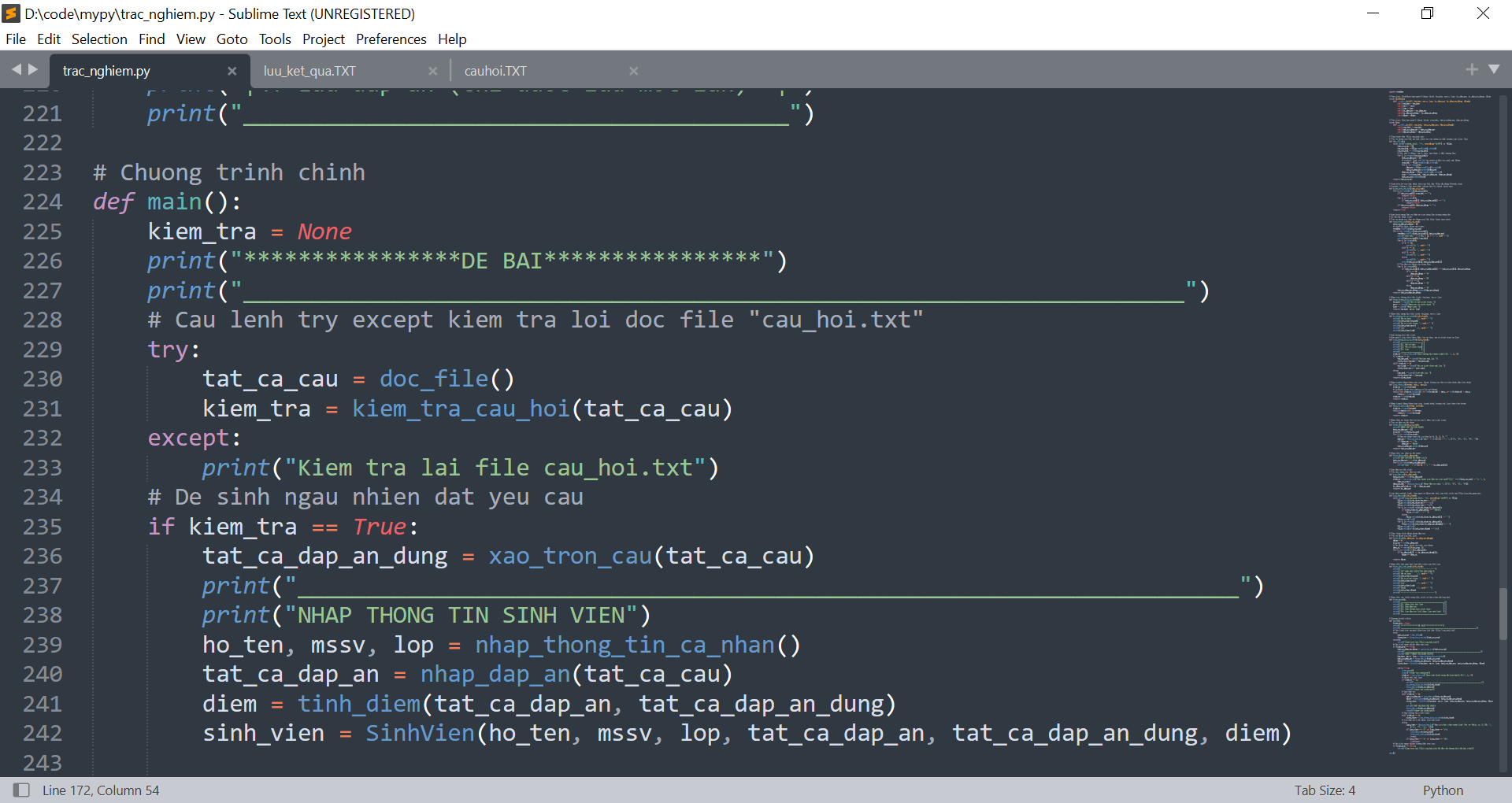
Những hàm đặc trưng của thư viện sẽ được comment.

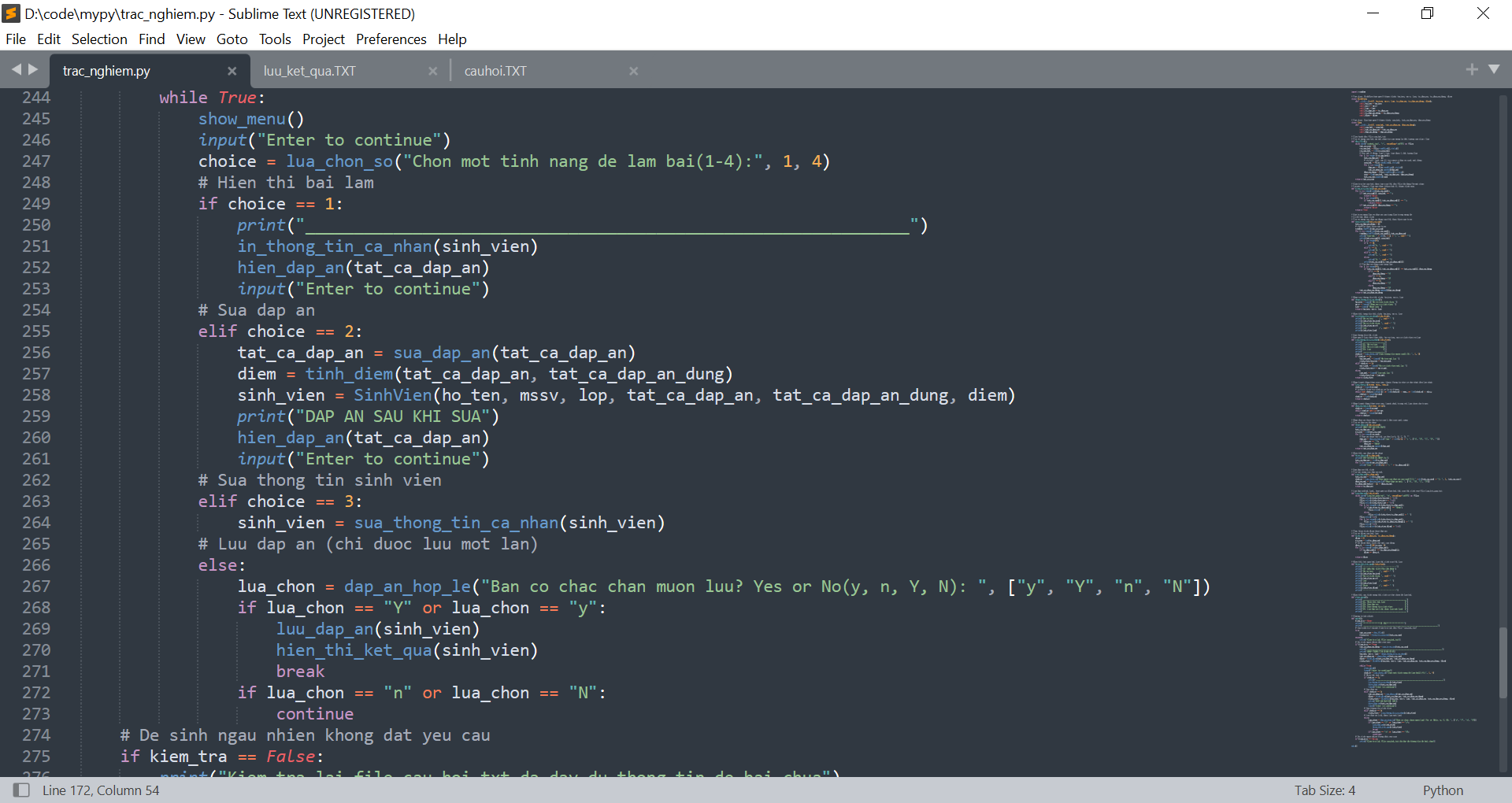


Comment chức năng của từng hàm và logic của chương trình.

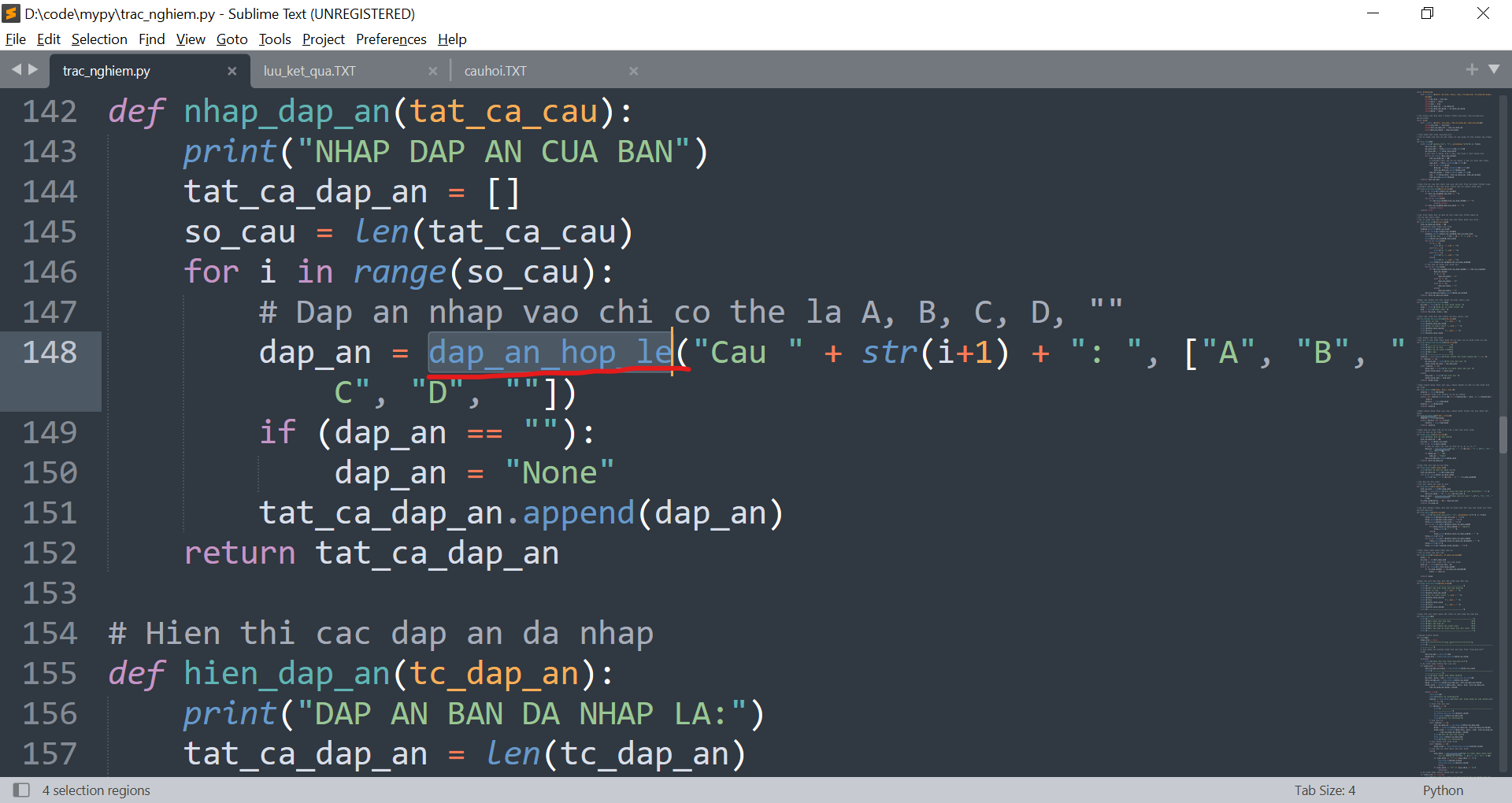


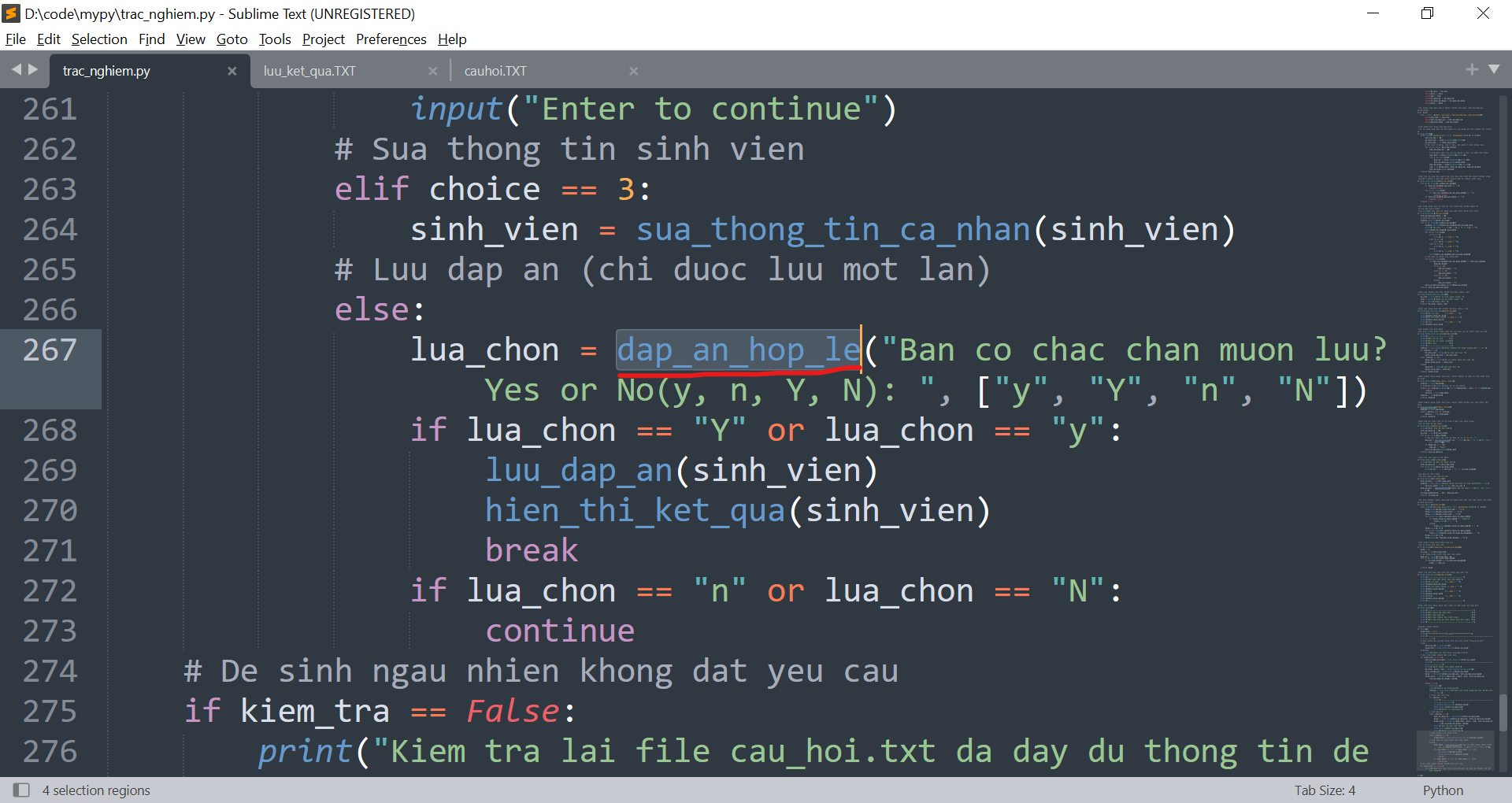
Chương trình chính được tổ chức thành các hàm, thủ tục, do đó, các bước của chương trình chính được thể hiện một cách tường minh, dễ nắm bắt, dễ thay đổi.

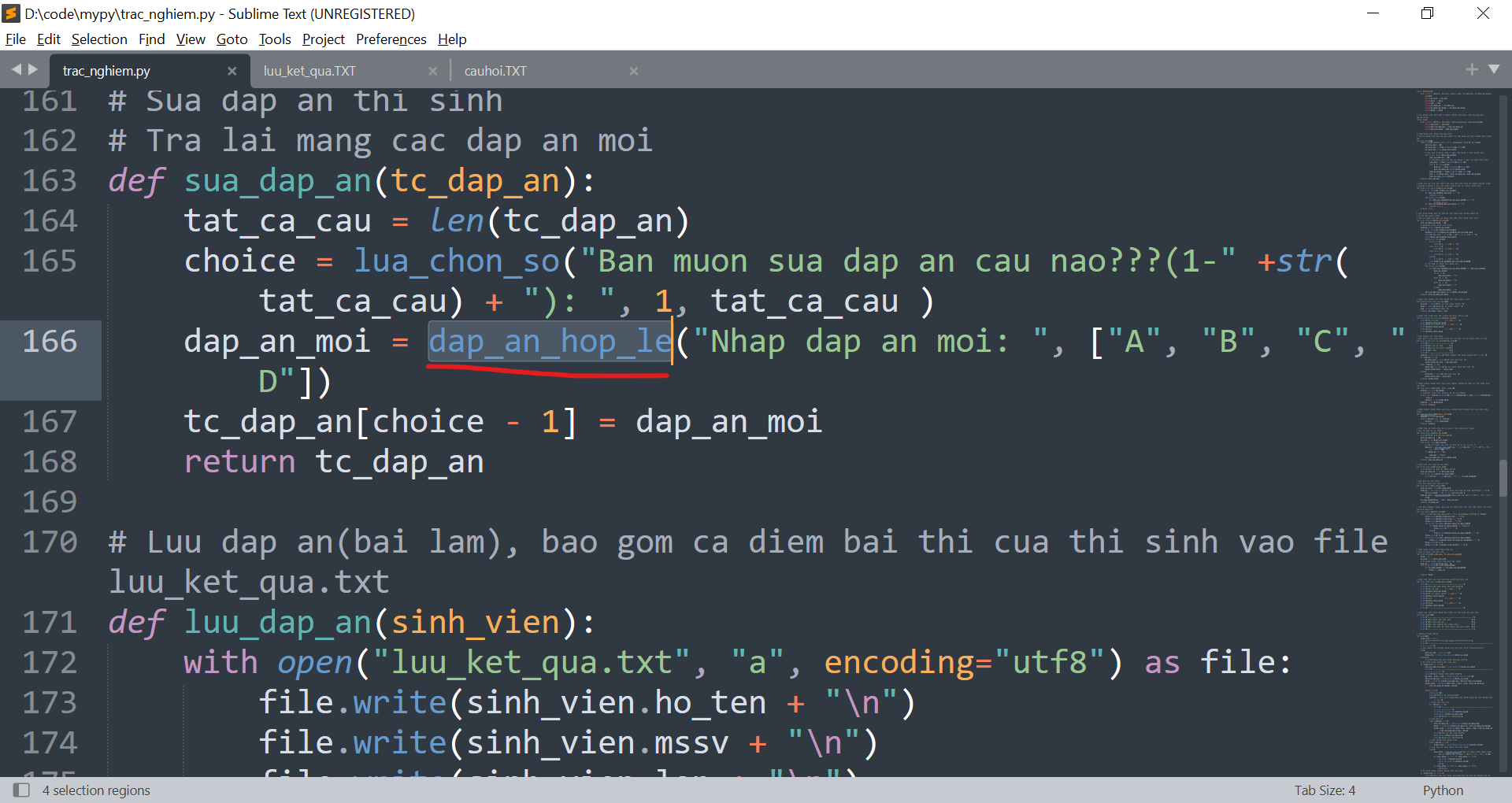


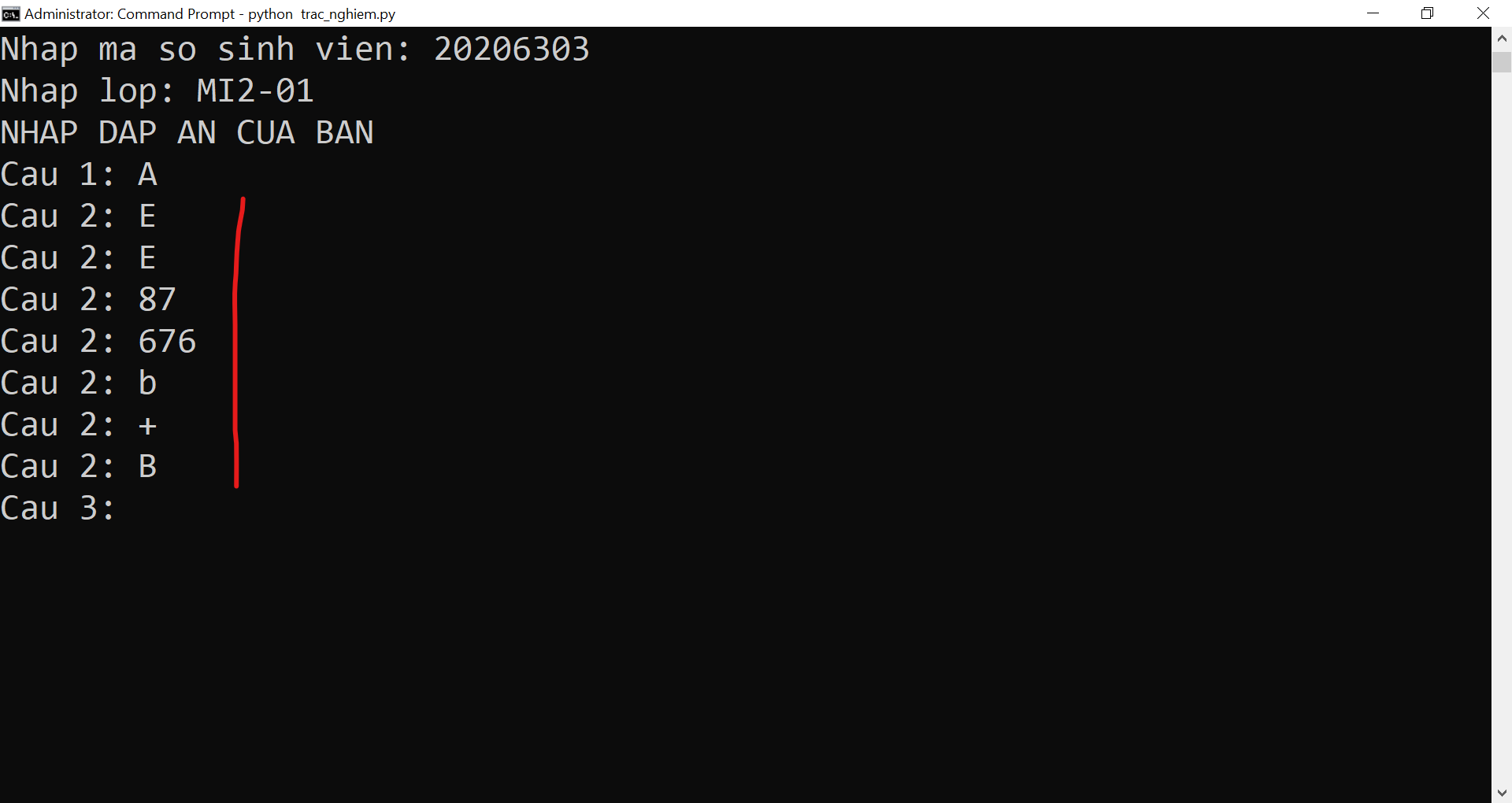


Hàm được tái sử dụng nhiều lần.







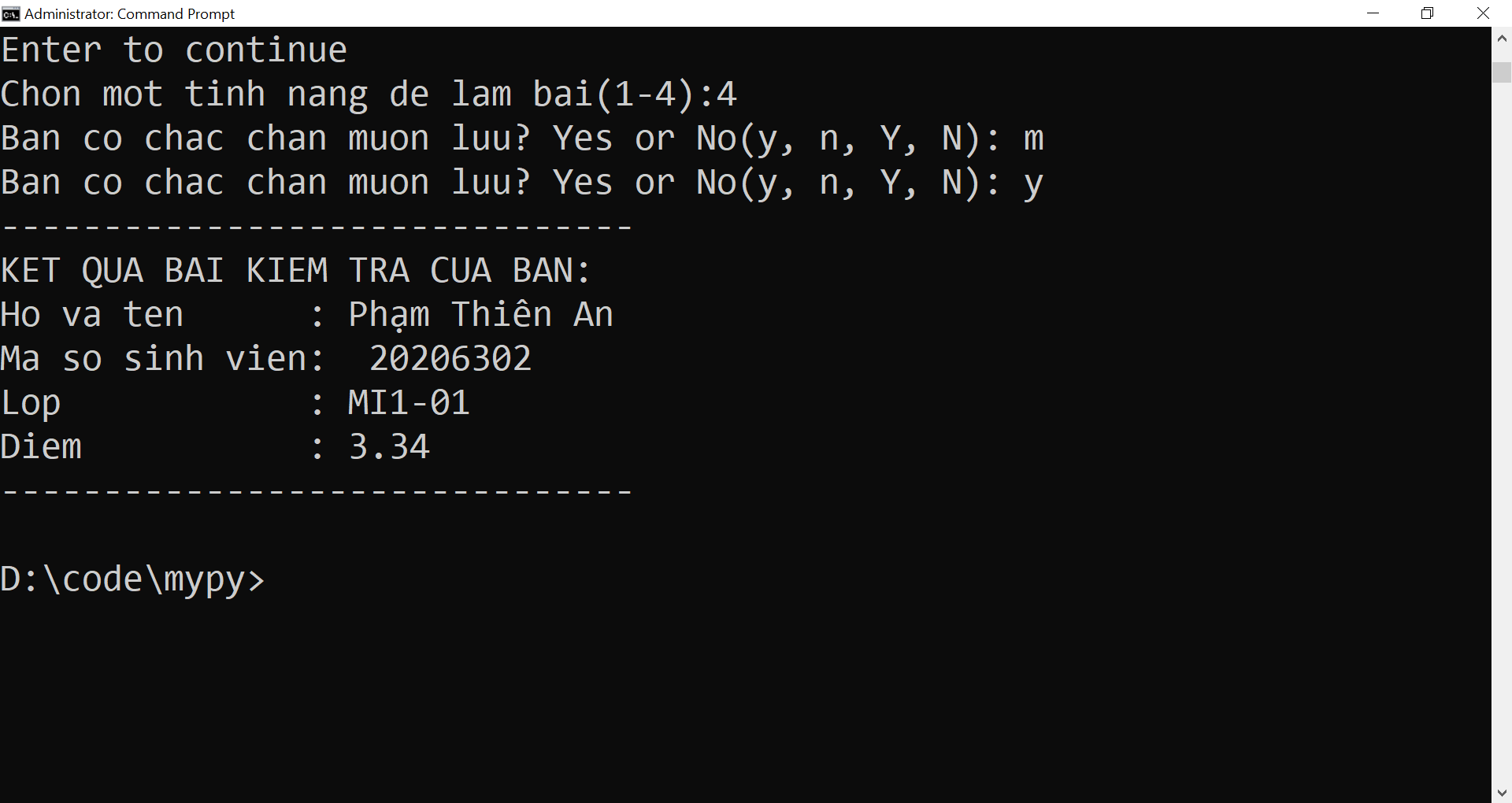
Do đây là một chương trình về quản lý nên cần phải phòng ngừa các input không mong muốn. Tránh các input không mong muốn bằng việc sử dụng hàm kiểm tra input. 

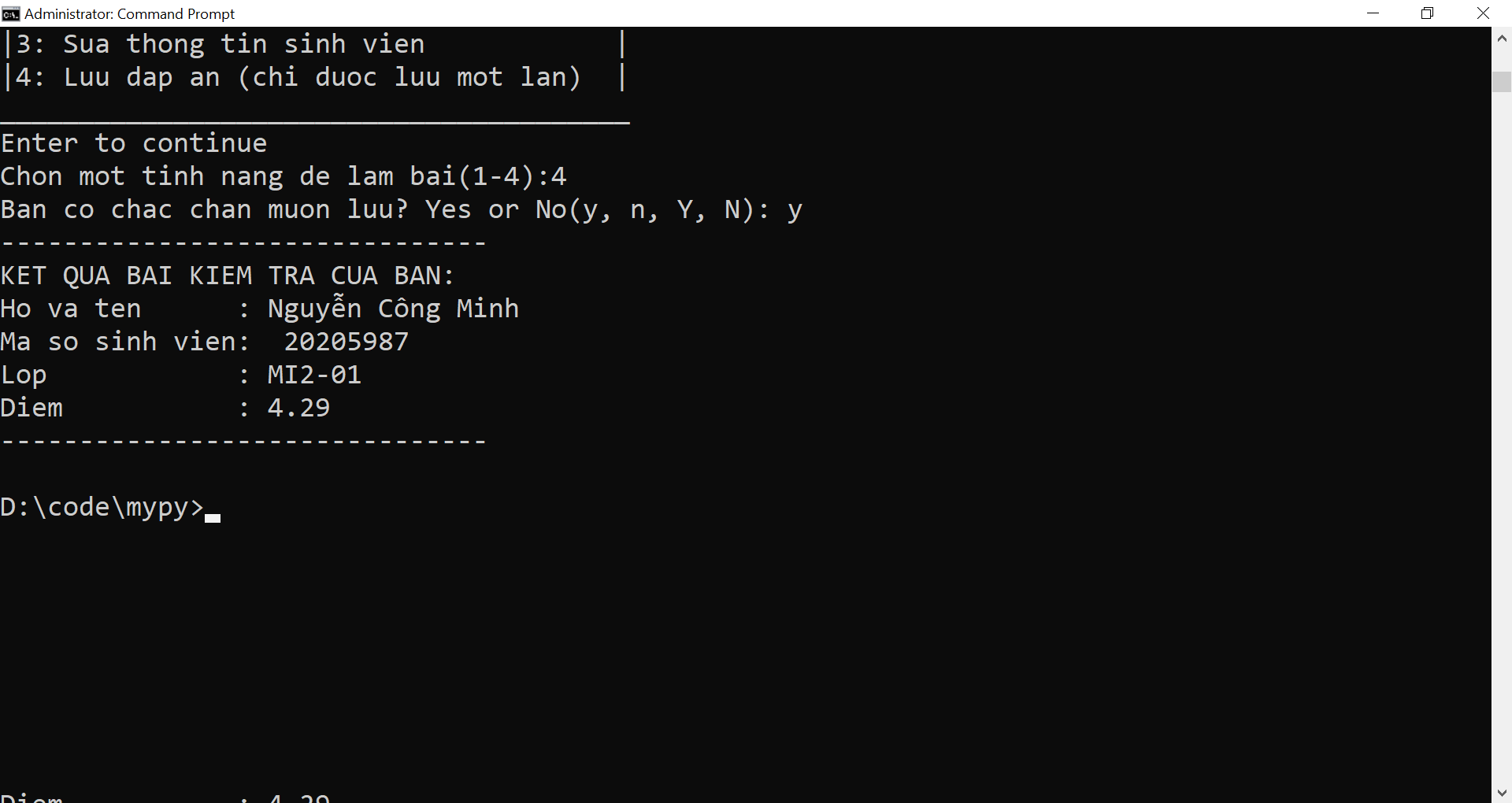
*Chỉ cho nhập A, B, C hoặc D*

Các hàm được xây dựng cần đảm bảo tính độc lập càng cao càng tốt, thay đổi trên hàm này không làm thay đổi hoặc ít thay đổi đến các hàm còn lại.

Xây dựng các hàm quan tâm đến tính trừu tượng hóa.

Bộ câu hỏi có tính trừu tượng và linh hoạt, khi thay đổi số câu hỏi trong một đề, đề bài cũng như cách tính điểm tự động thay đổi theo.

*Điểm của bài thi gồm 6 câu hỏi trắc nghiệm*



*Điểm của bài thi gồm 7 câu hỏi trắc nghiệm*

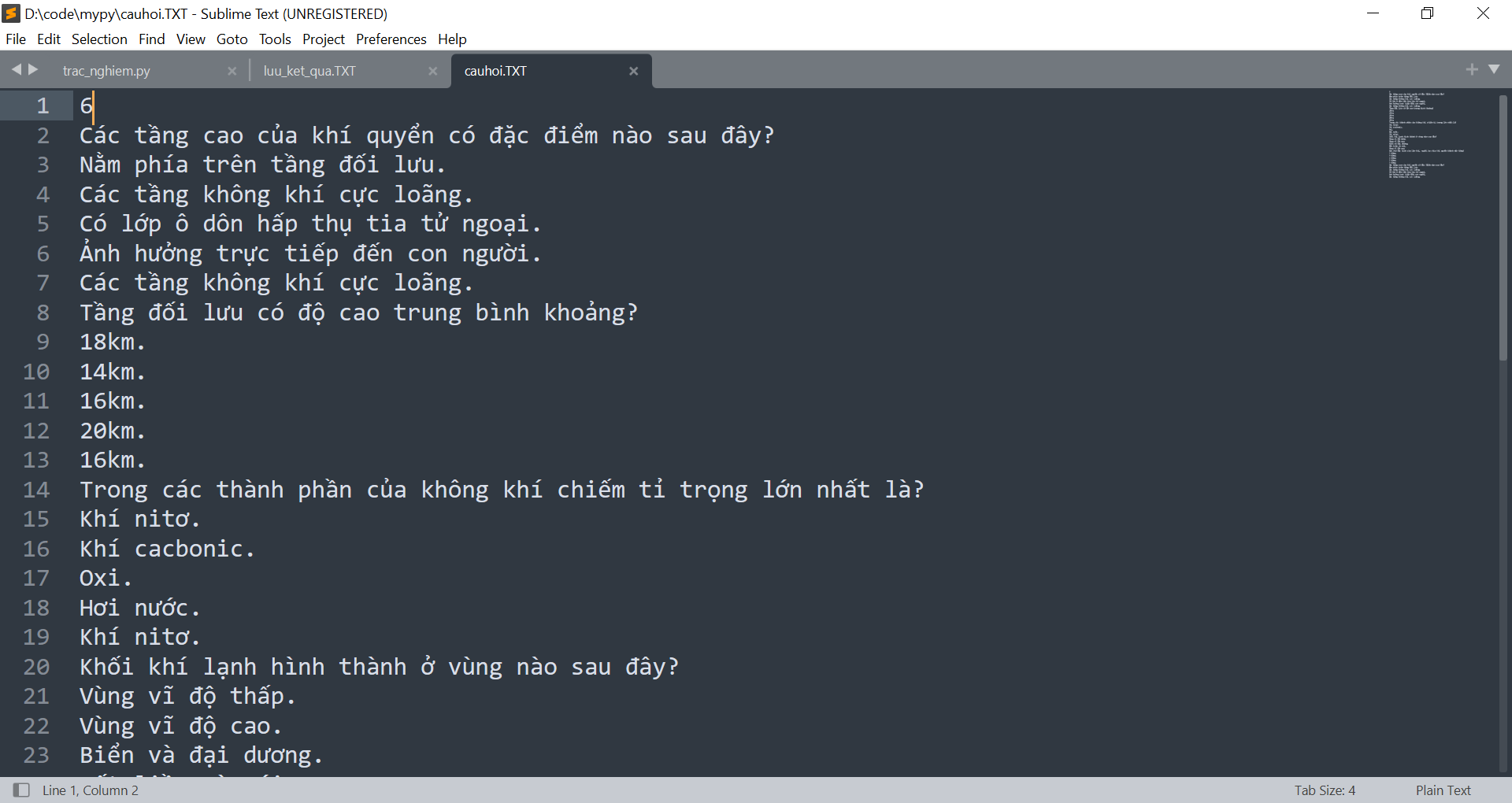
Kiểm tra lỗi:

Hàm try except và kiểm tra bộ câu hỏi đã đúng với yêu cầu chưa?

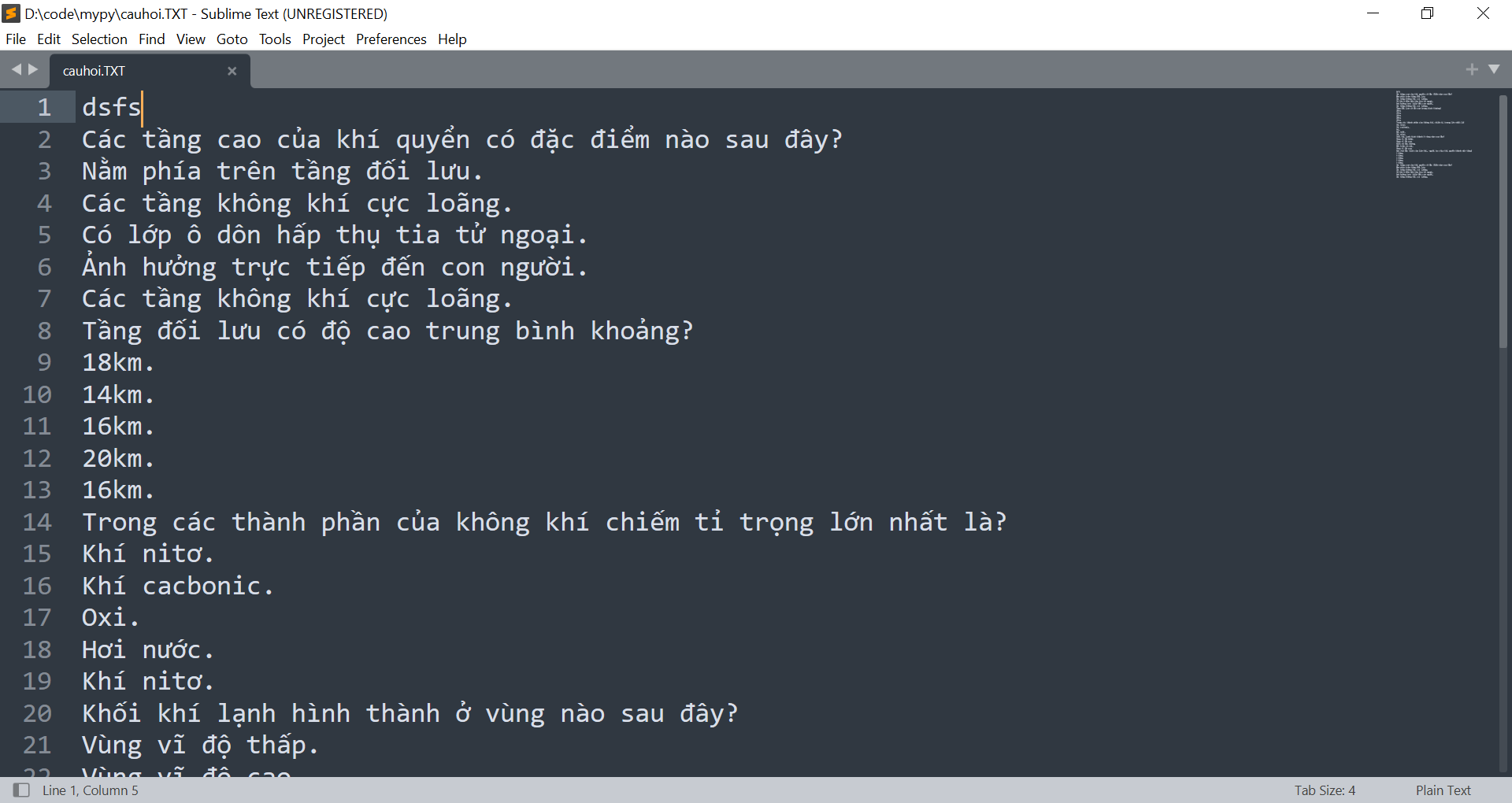
# KẾT QUẢ THỰC HIỆN CHƯƠNG TRÌNH

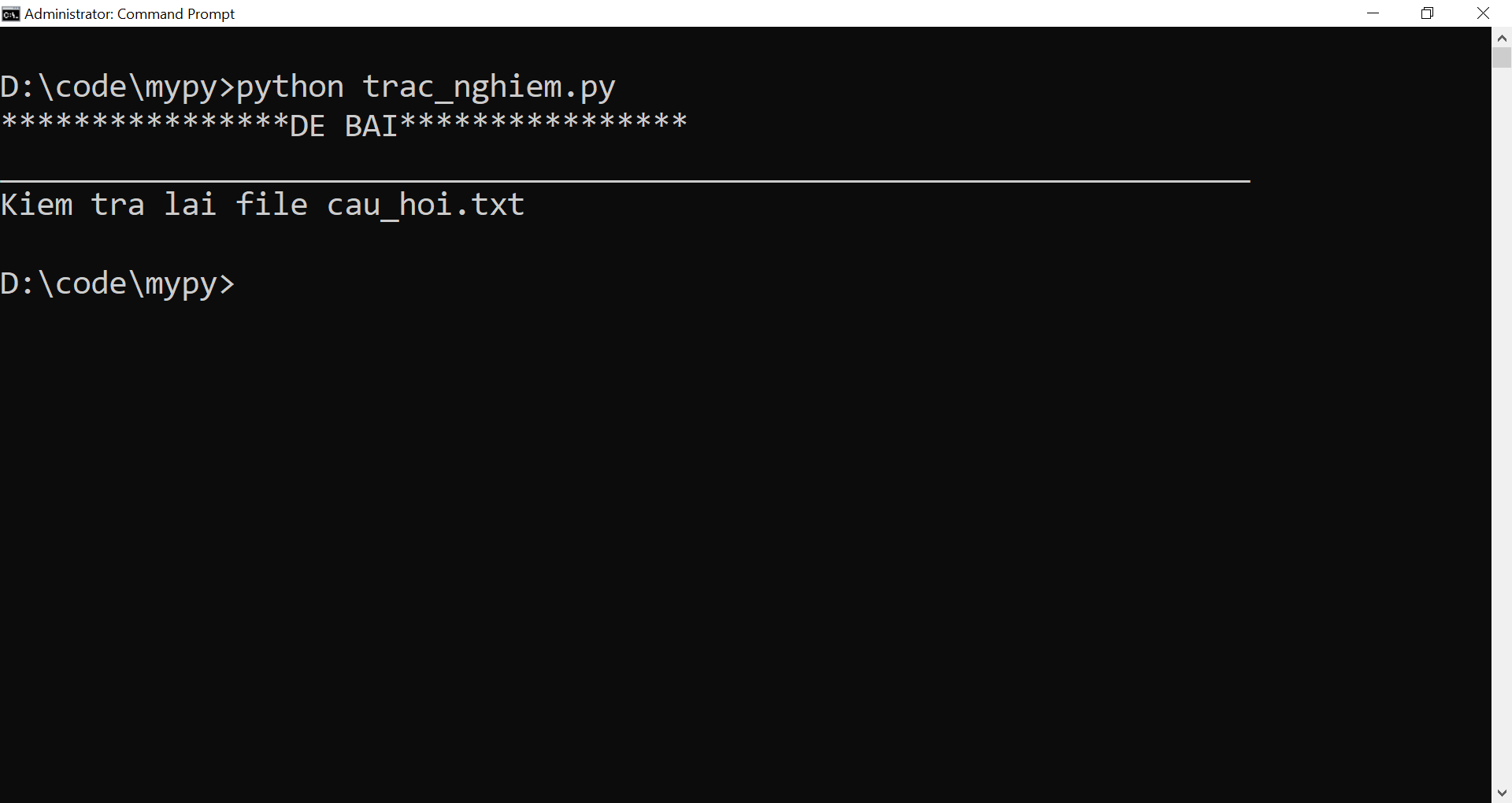
**Các dữ liệu kiểm thử được lưu tại file “input\_trac\_nghiem.txt”.**

File cau\_hoi.txt.



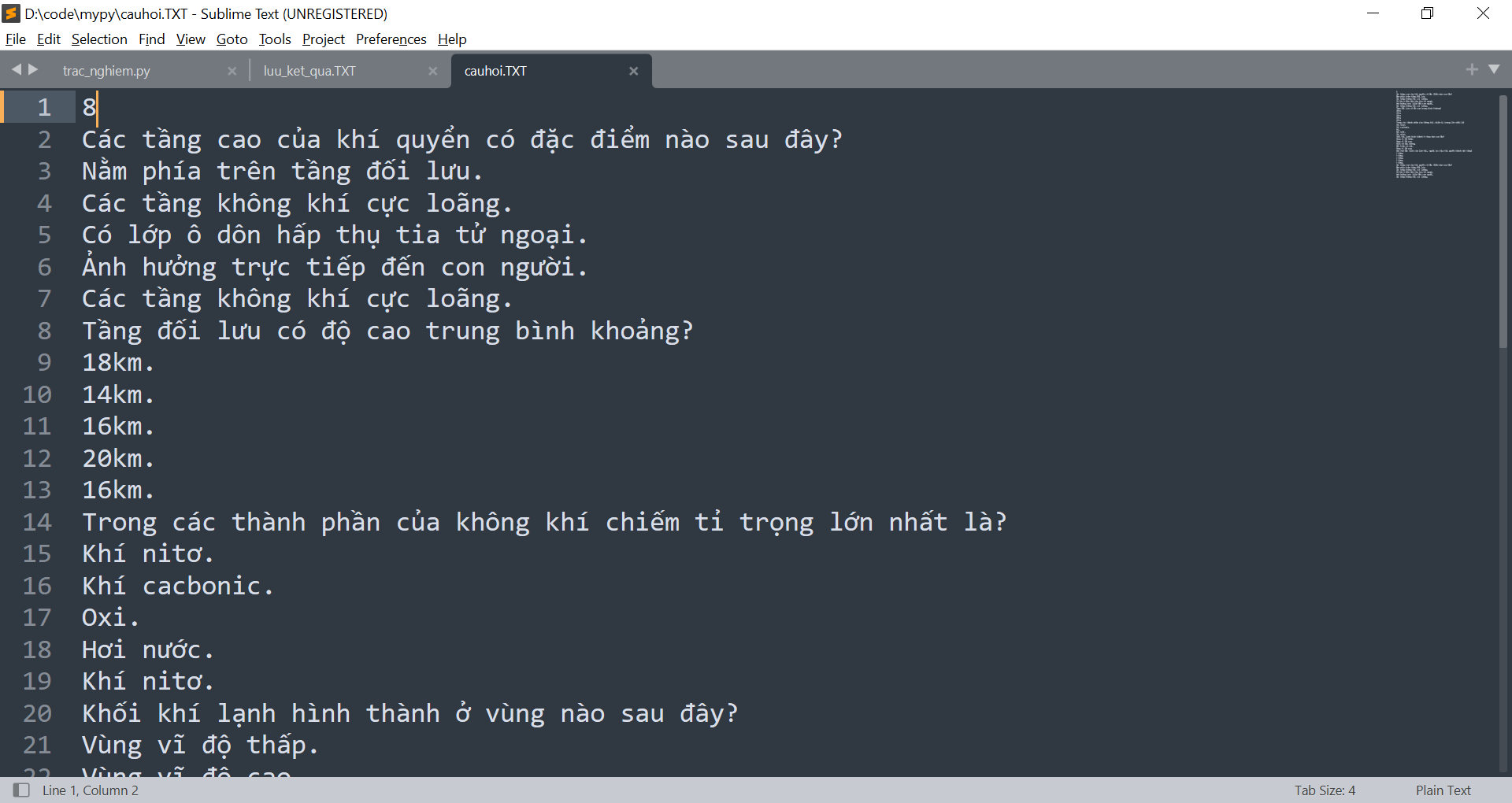
Trường hợp dòng đầu tiên của file cau\_hoi.txt sai định dạng.

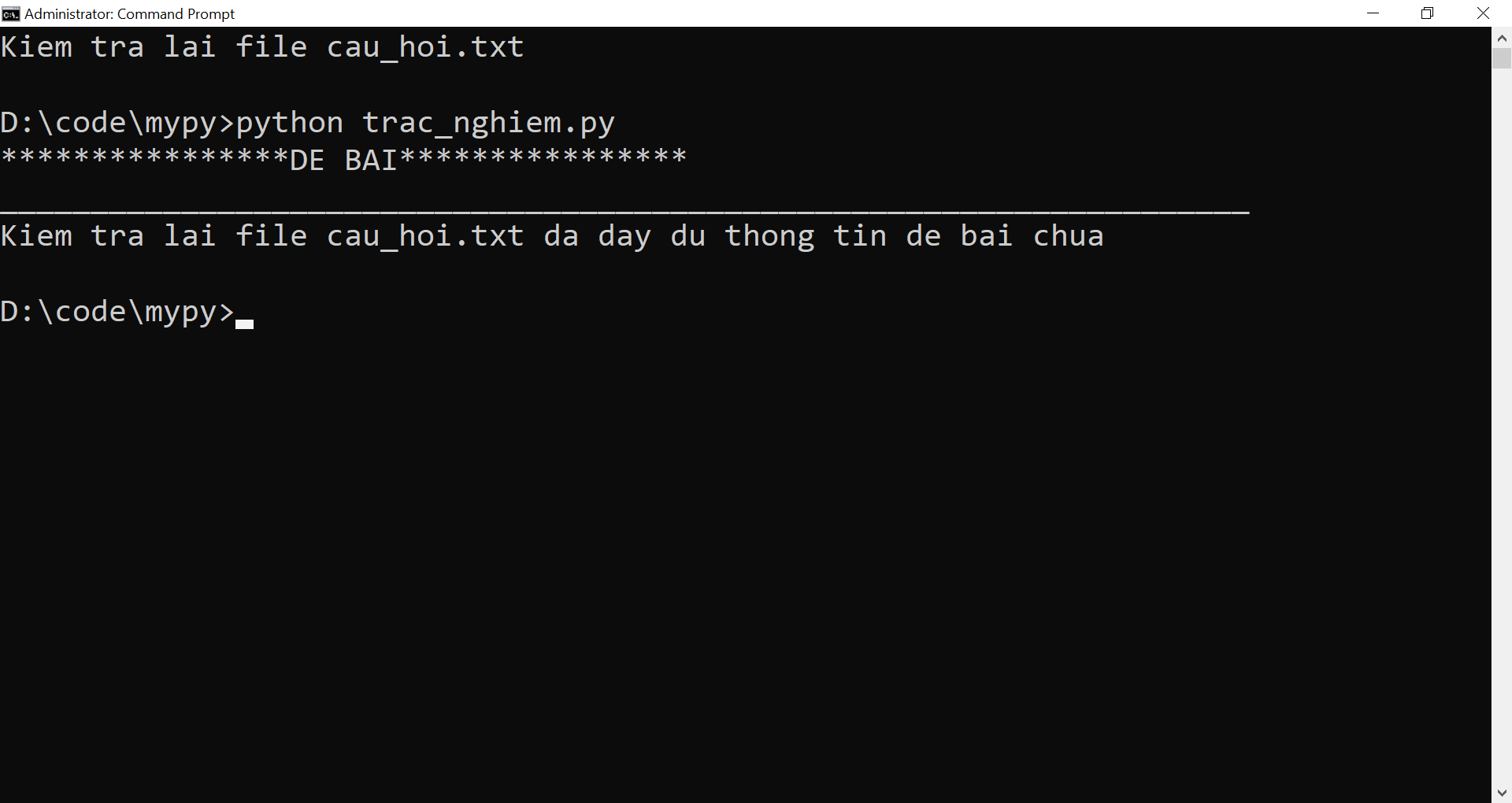




*Sử dụng hàm try except kiểm tra lỗi*

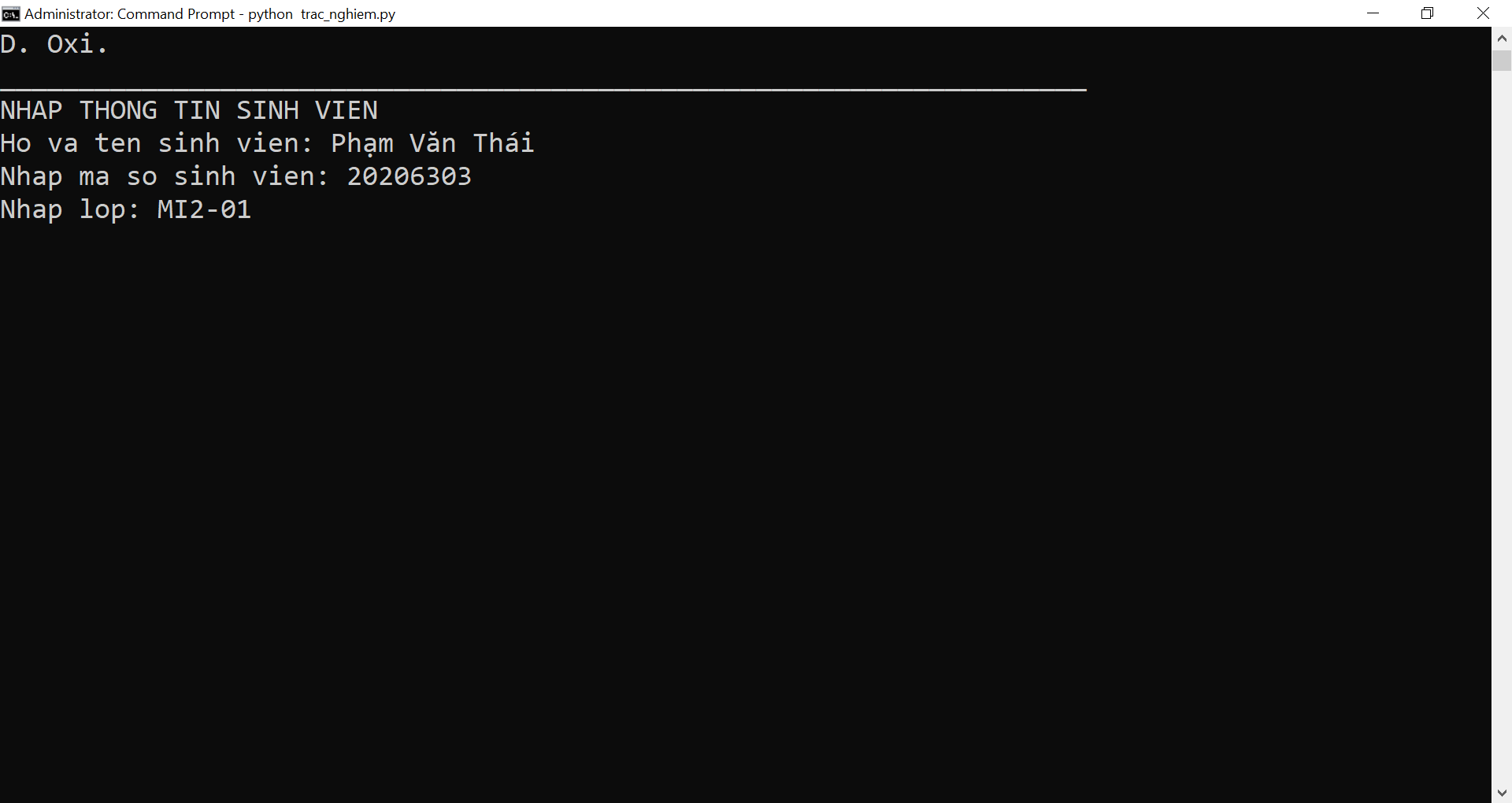
Trường hợp ghi 8 câu hỏi nhưng trong file chỉ có 6 câu hỏi.





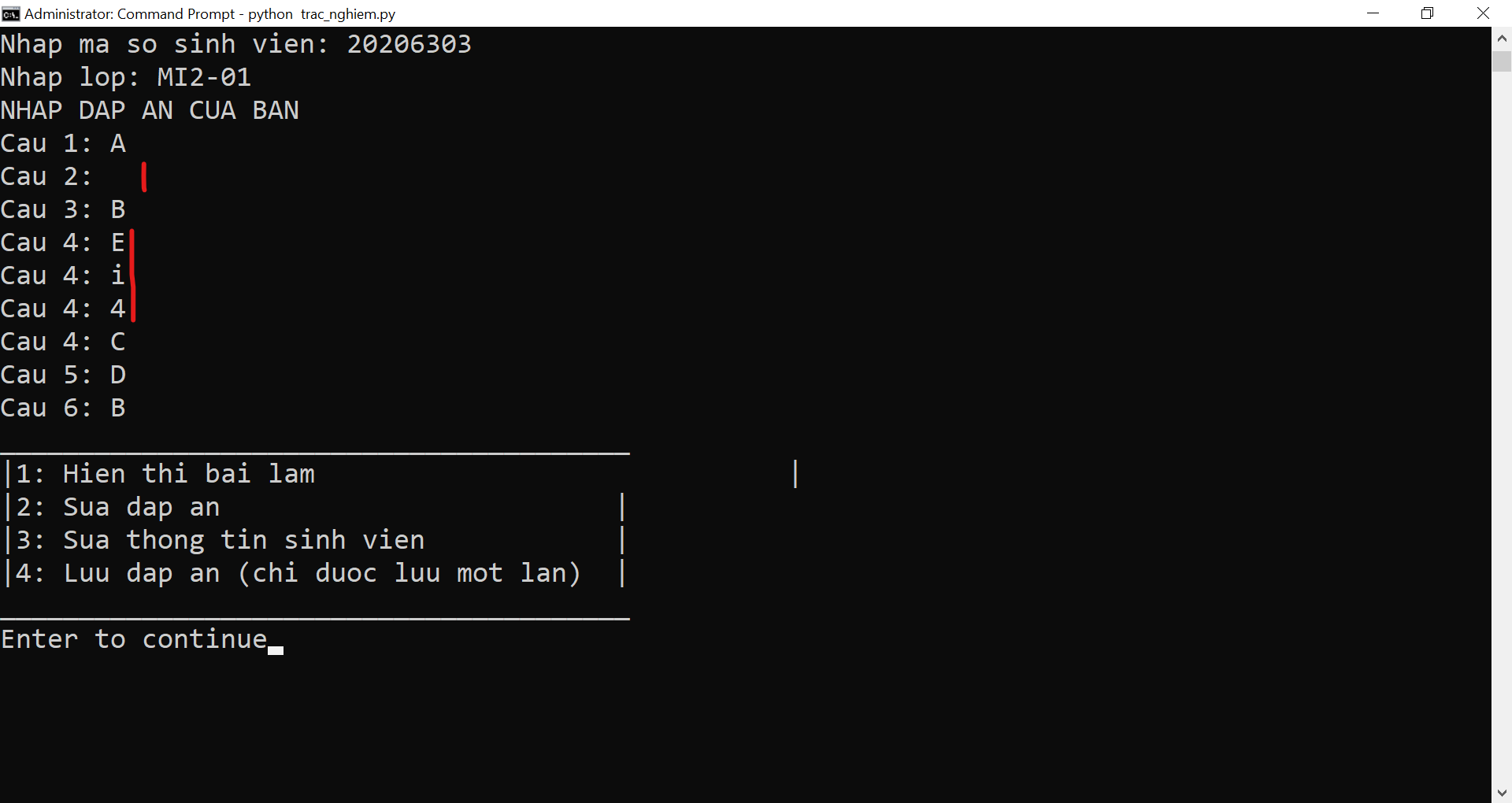
**Trường hợp đề bài gồm 6 câu hỏi:**

Điền thông tin thí sinh bao gồm họ và tên, mã số sinh viên và lớp.

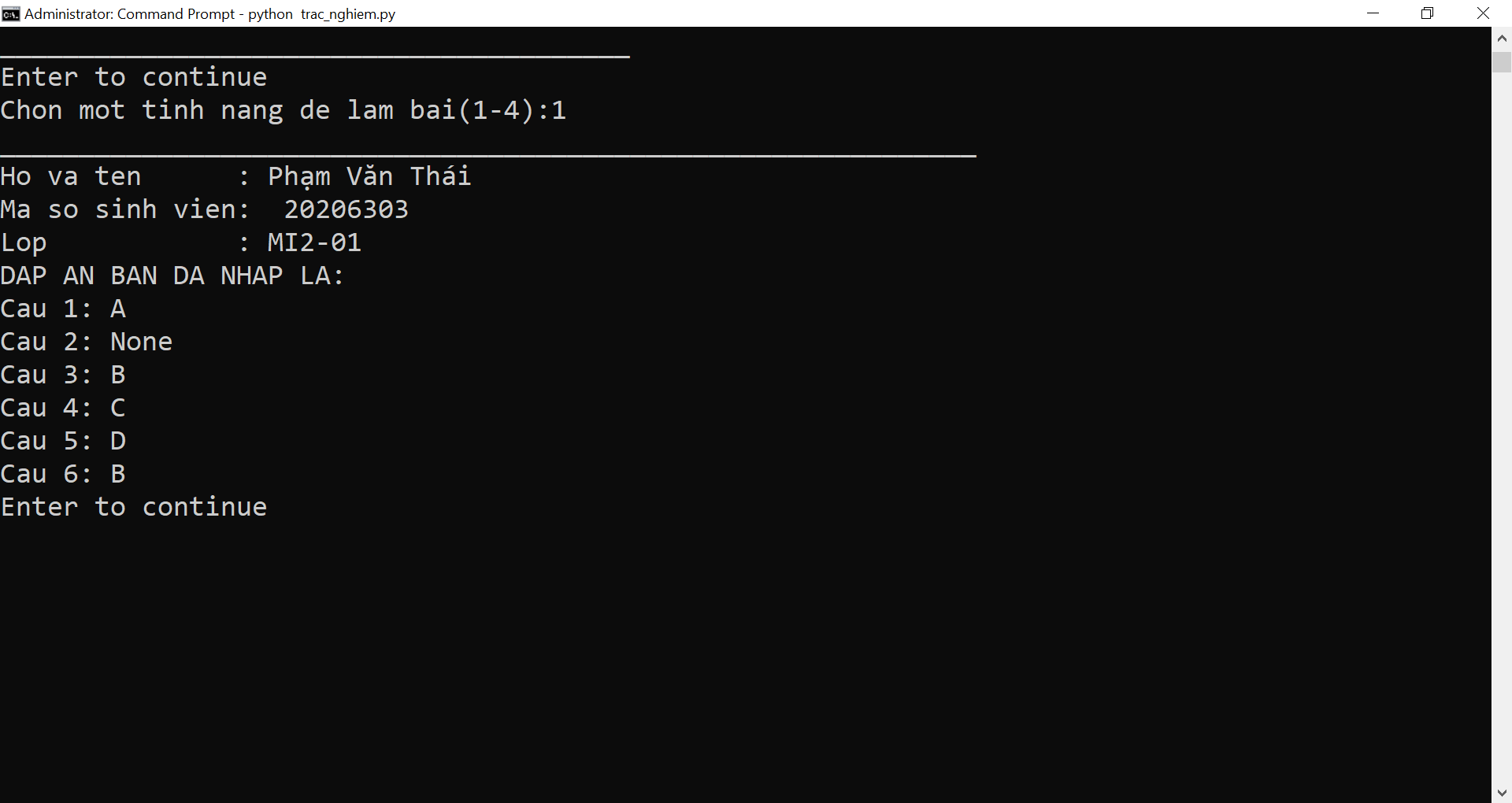


Nhập đáp án trả lời của thí sinh (chấp nhận trường hợp bỏ không điền).

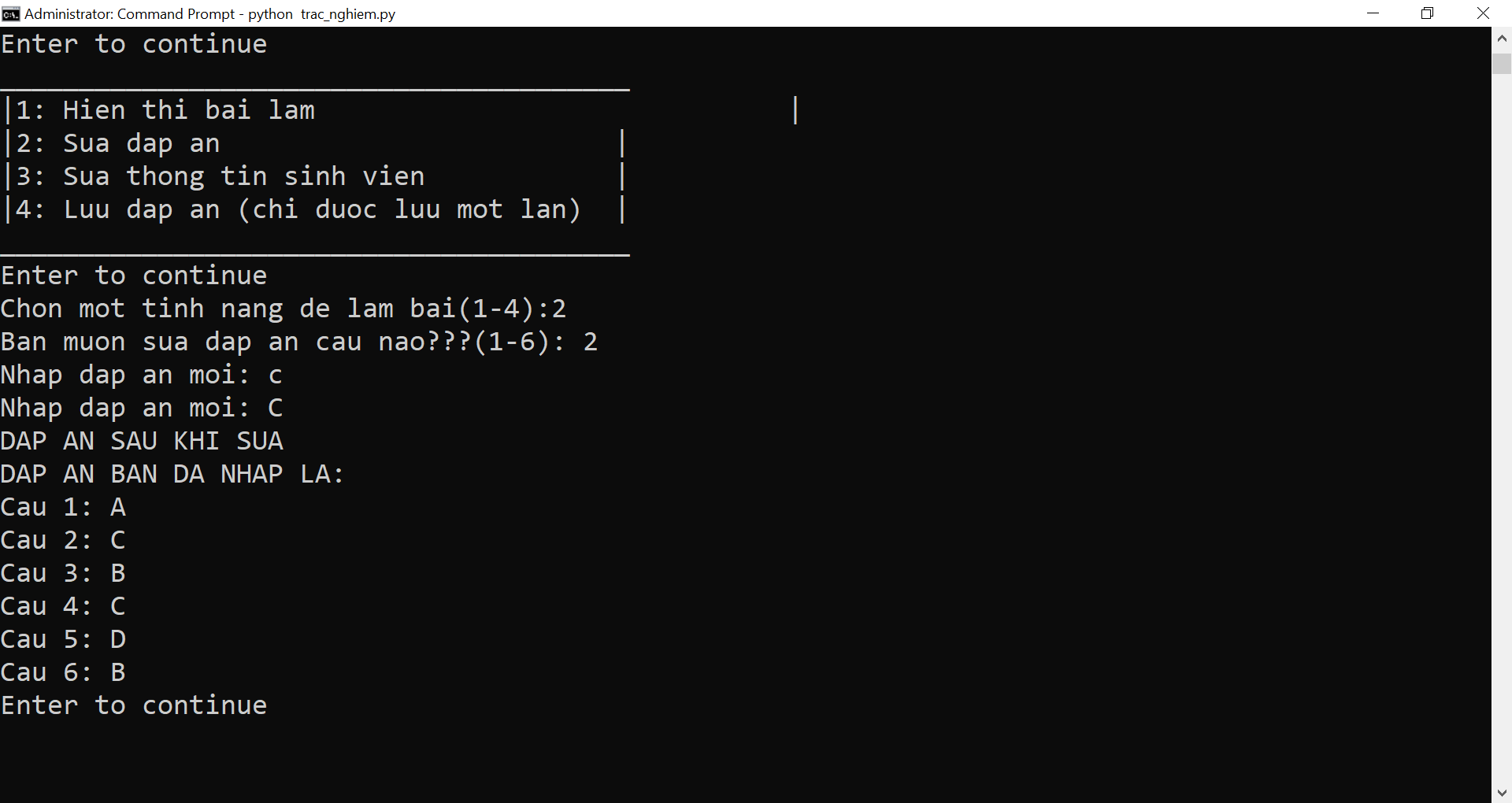
Trường hợp nhập sai đáp án và trường hợp nhập đúng đáp án.



Hiển thị bài làm của thí sinh, đáp án không điền được hiển thị là None.

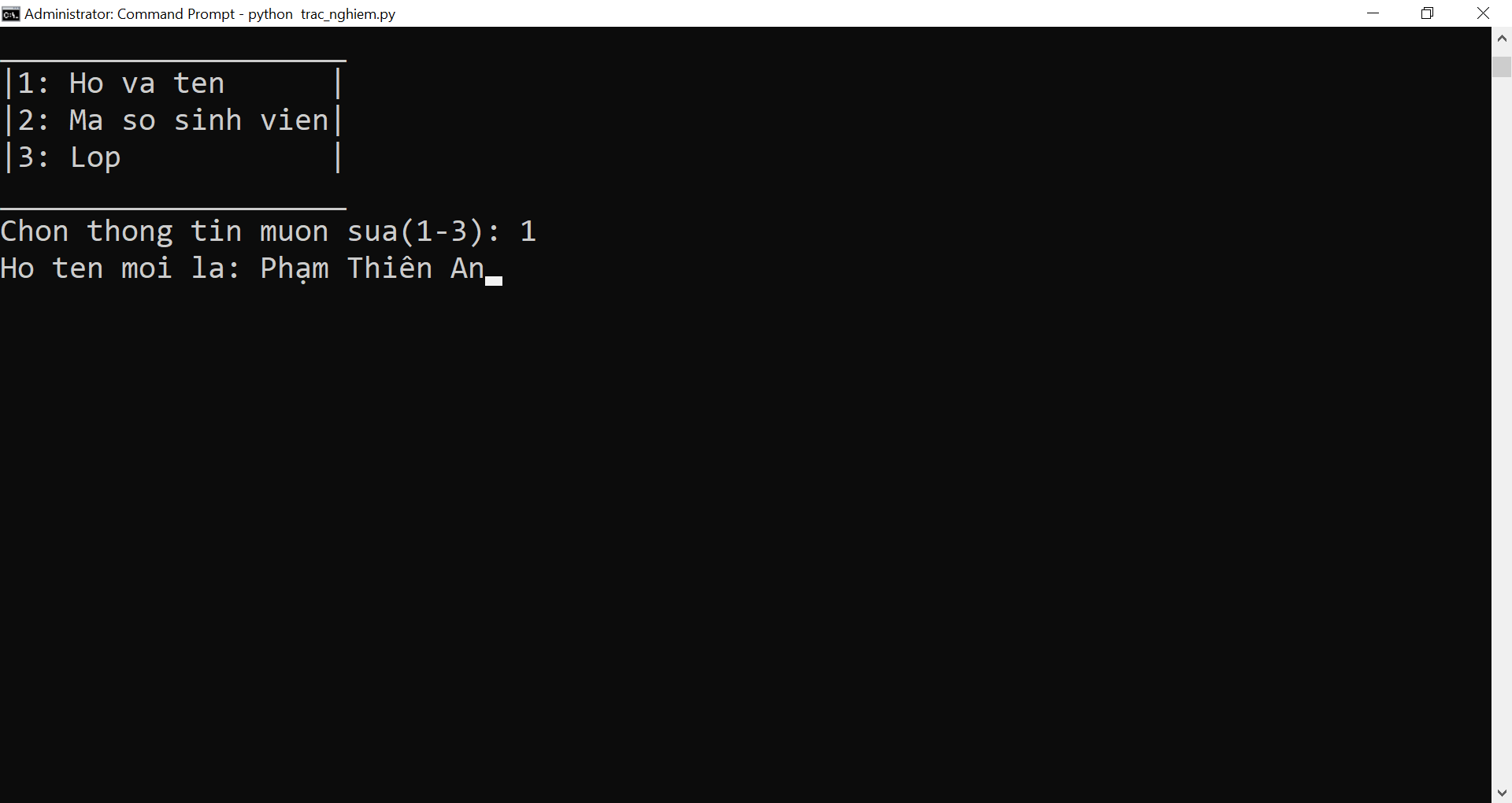


Thực hiện sửa đáp án và hiển thị đáp án sau khi sửa.(Không chấp nhận không điền).

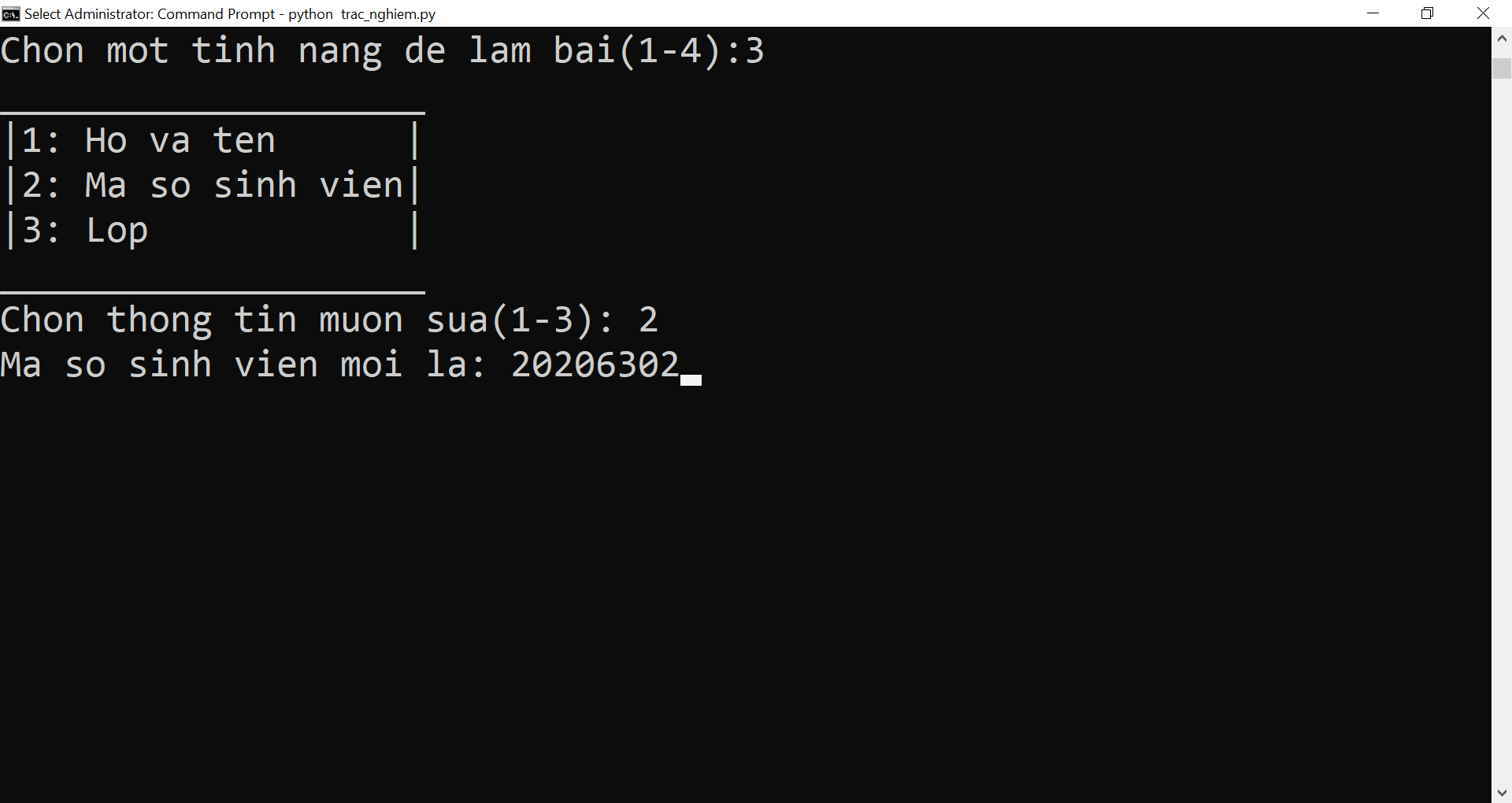


Sửa thông tin của thí sinh:

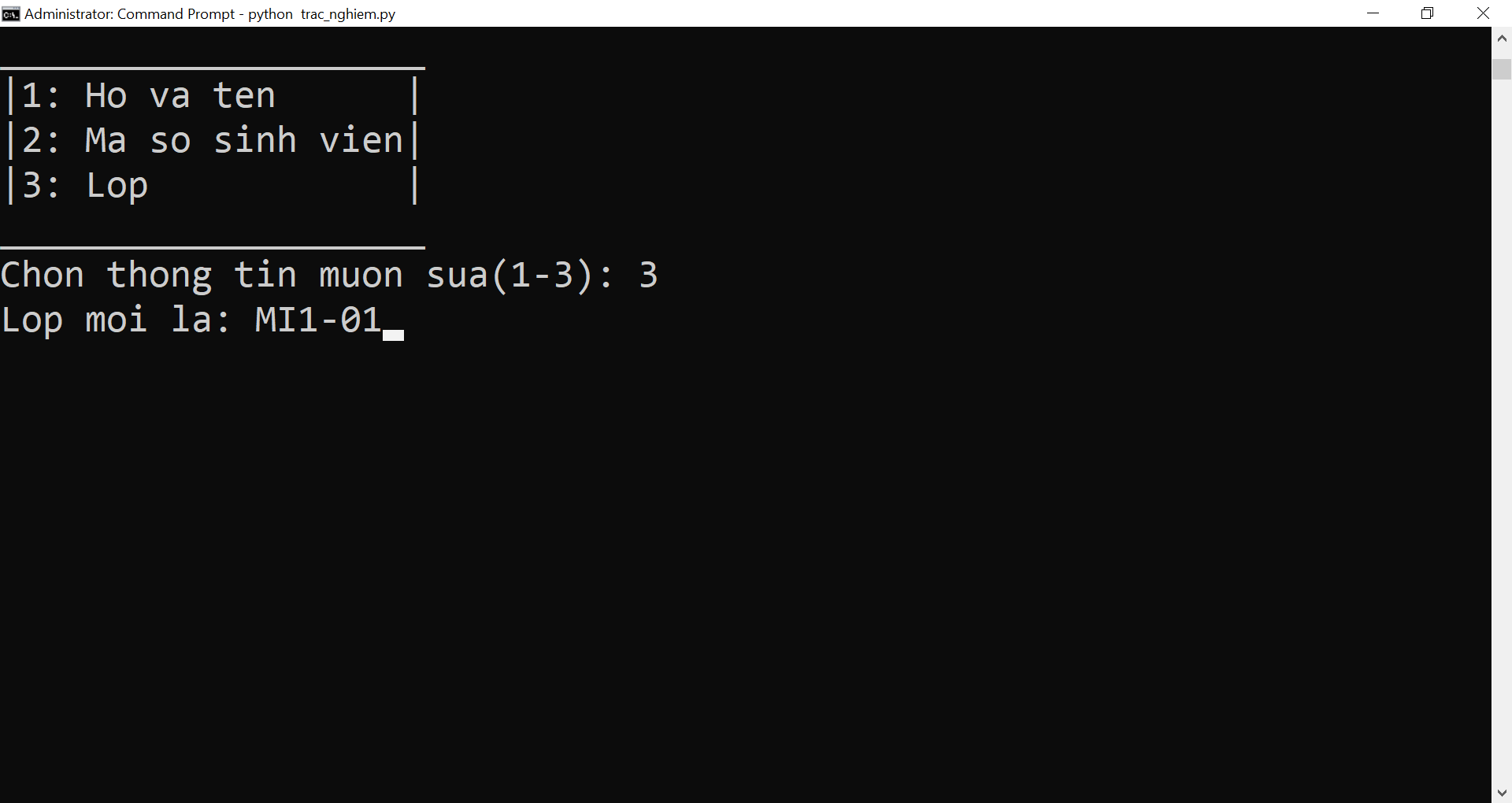
Sửa họ và tên thí sinh:



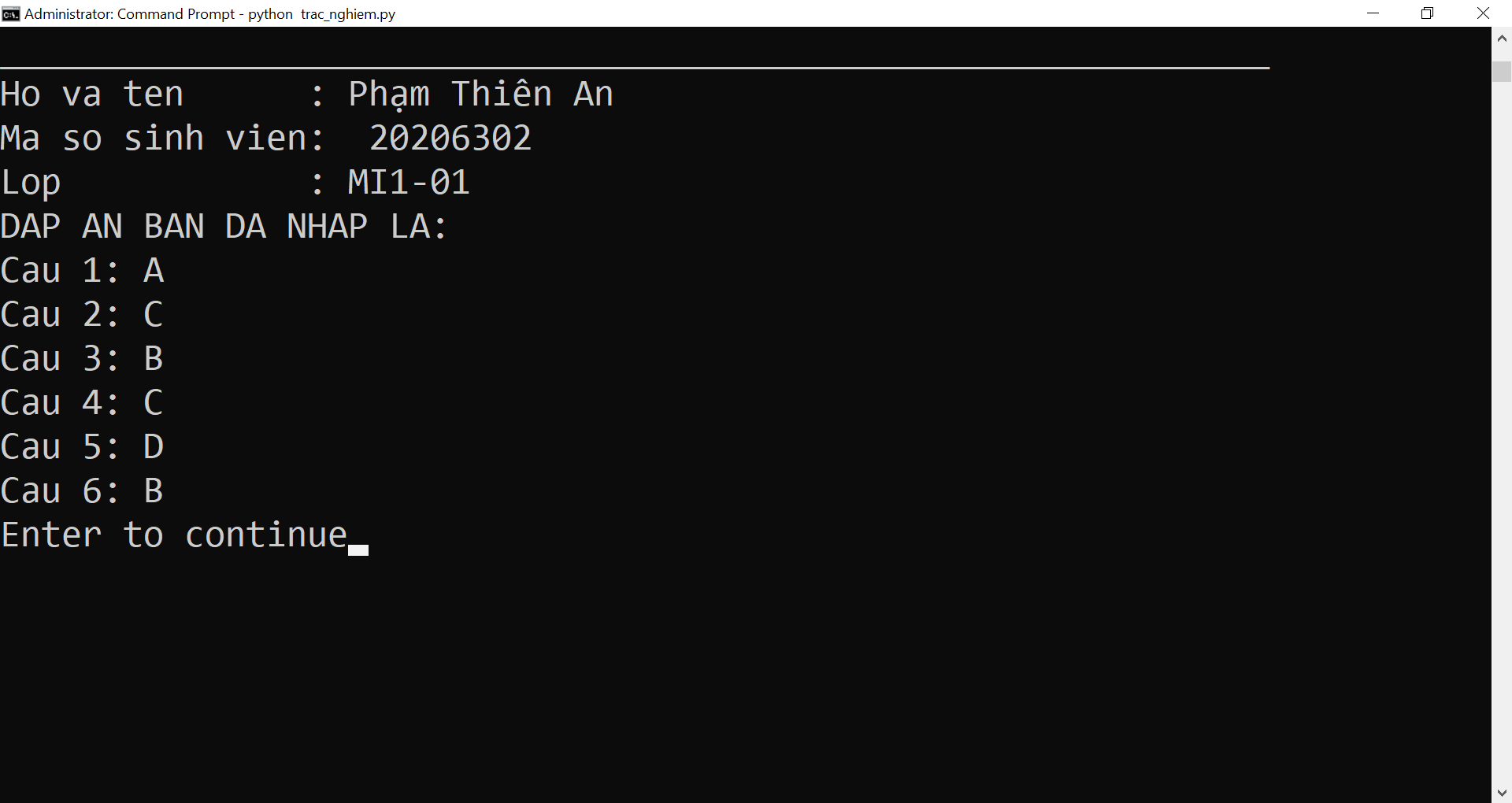
Sửa mã số sinh viên:



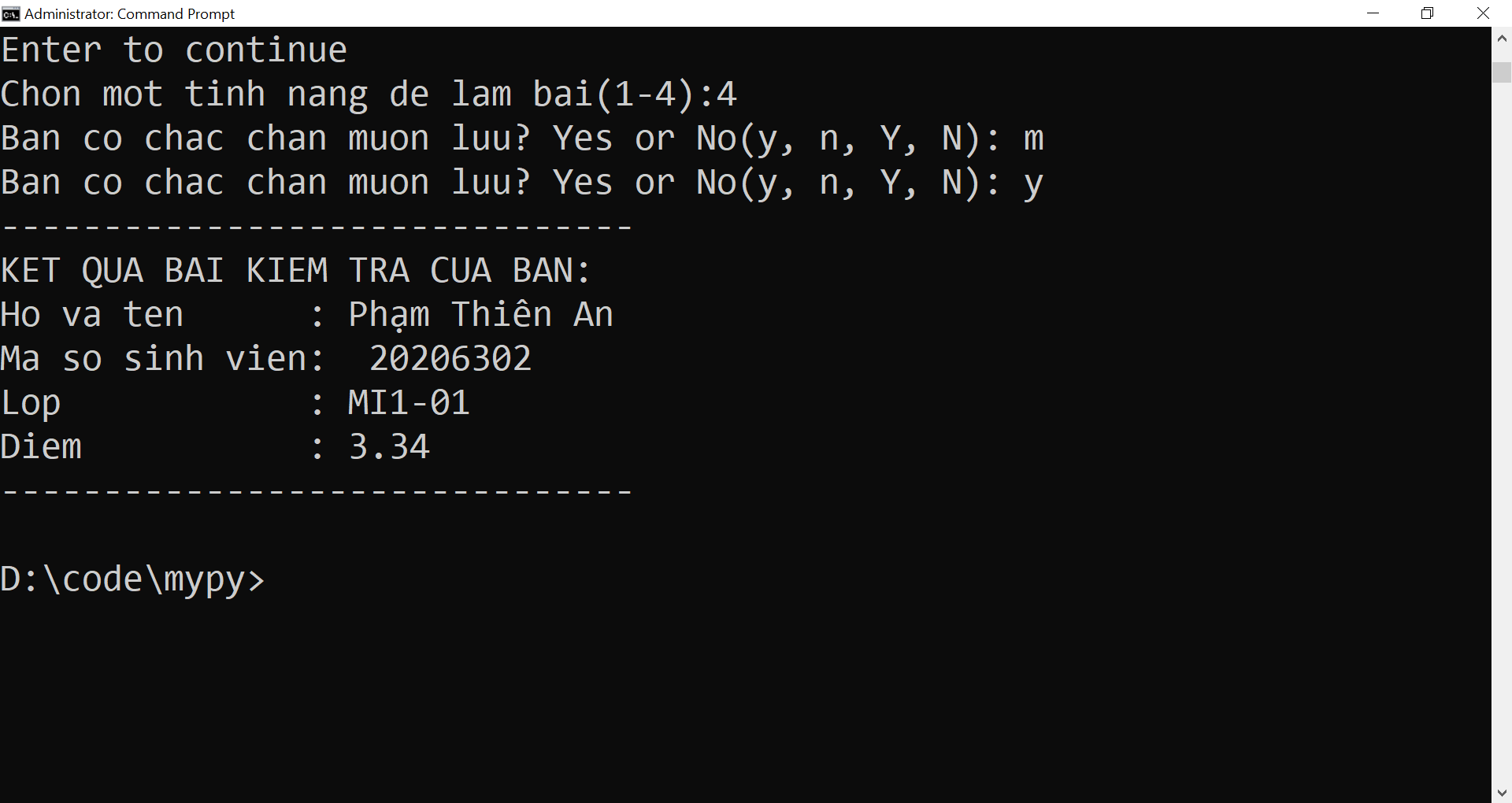
Sửa lớp thí sinh:



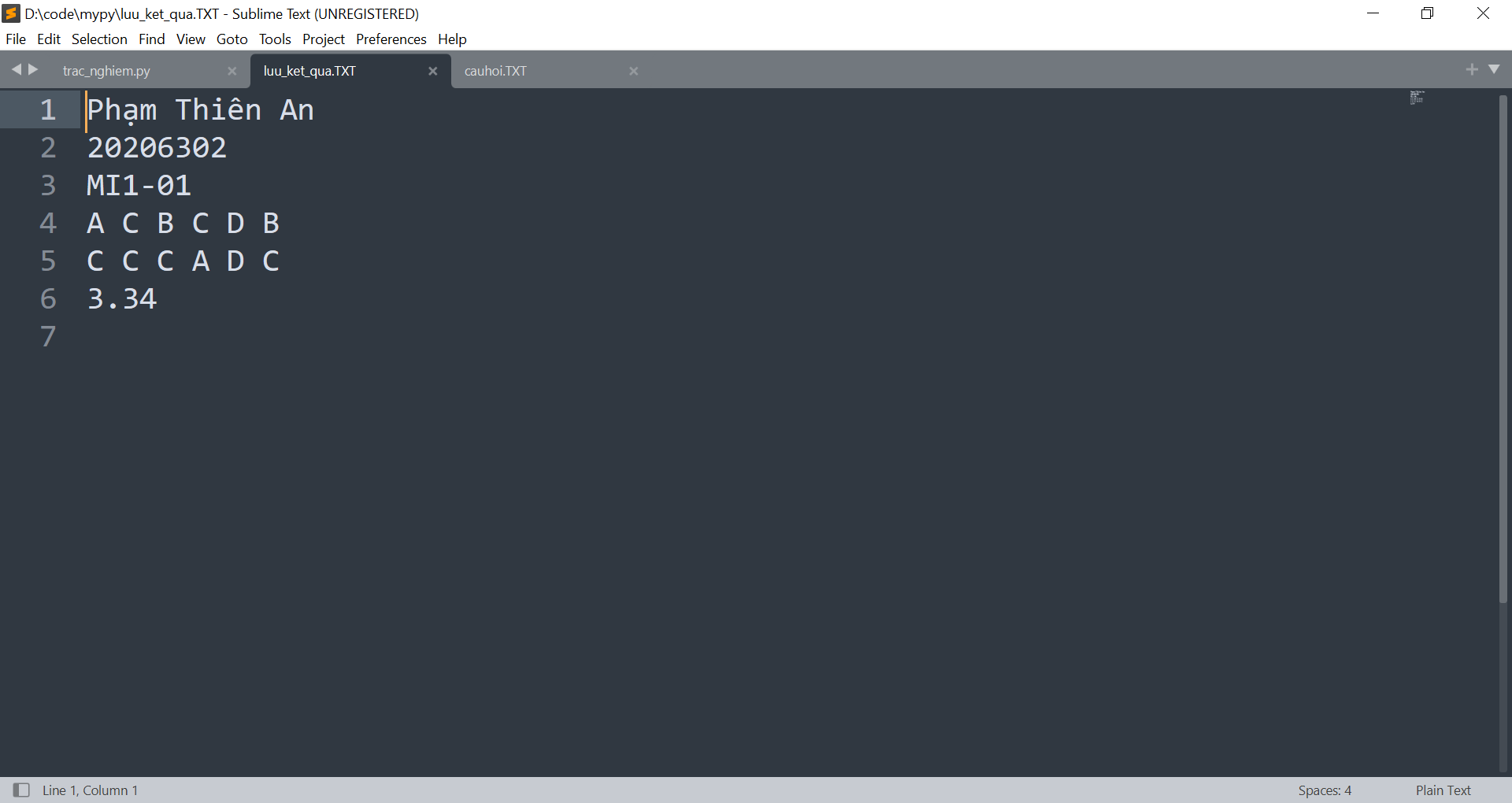
Hiển thị lại thông tin bài làm sau khi thay đổi.



Lưu thông tin và hiển thị bài làm thí sinh.

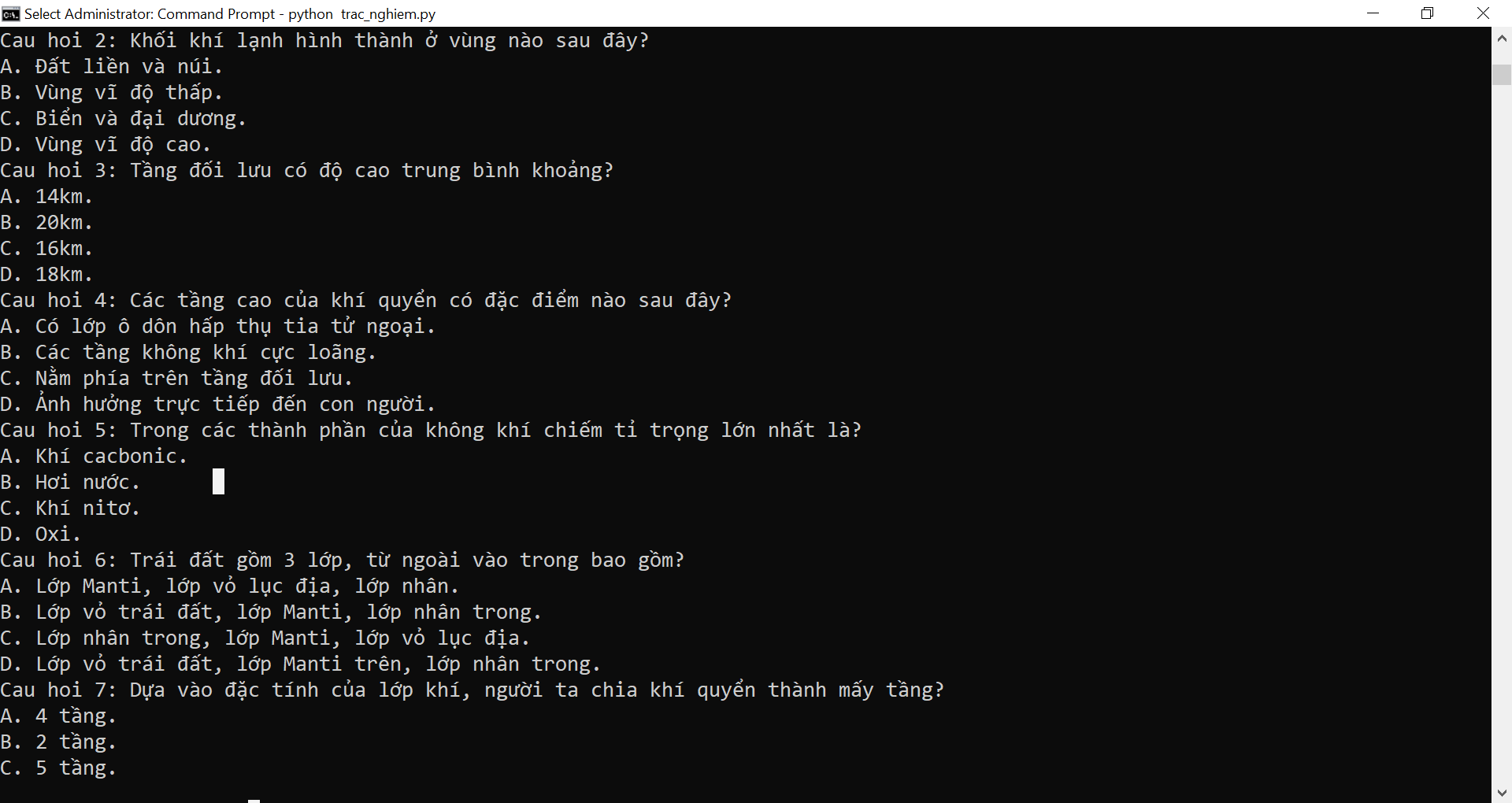


File lưu kết quả bài làm thí sinh.



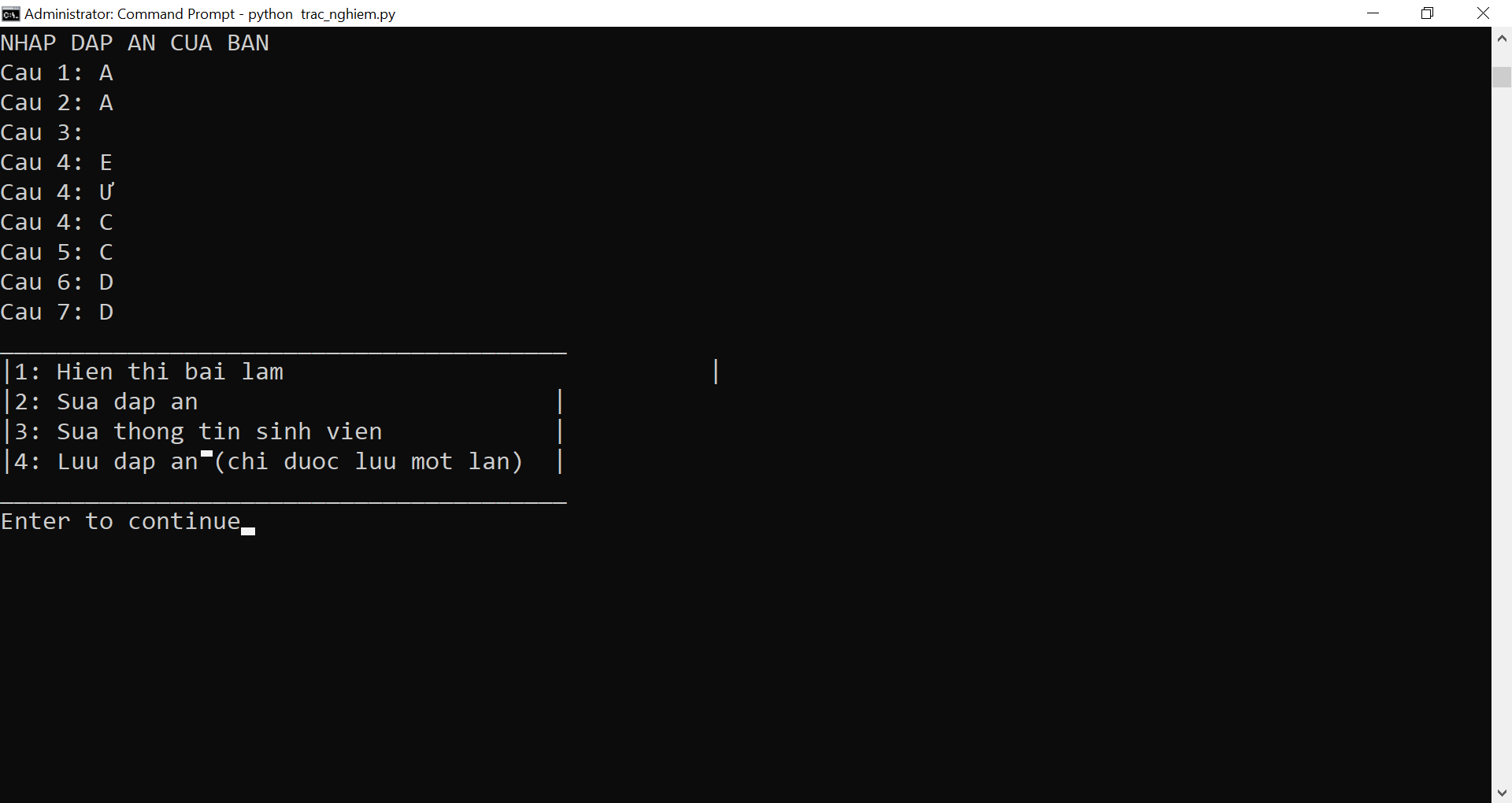
**Trường hợp đề bài gồm 7 câu hỏi:**

Hiển thị đề bài trắc nghiệm được sinh ngẫu nhiên.

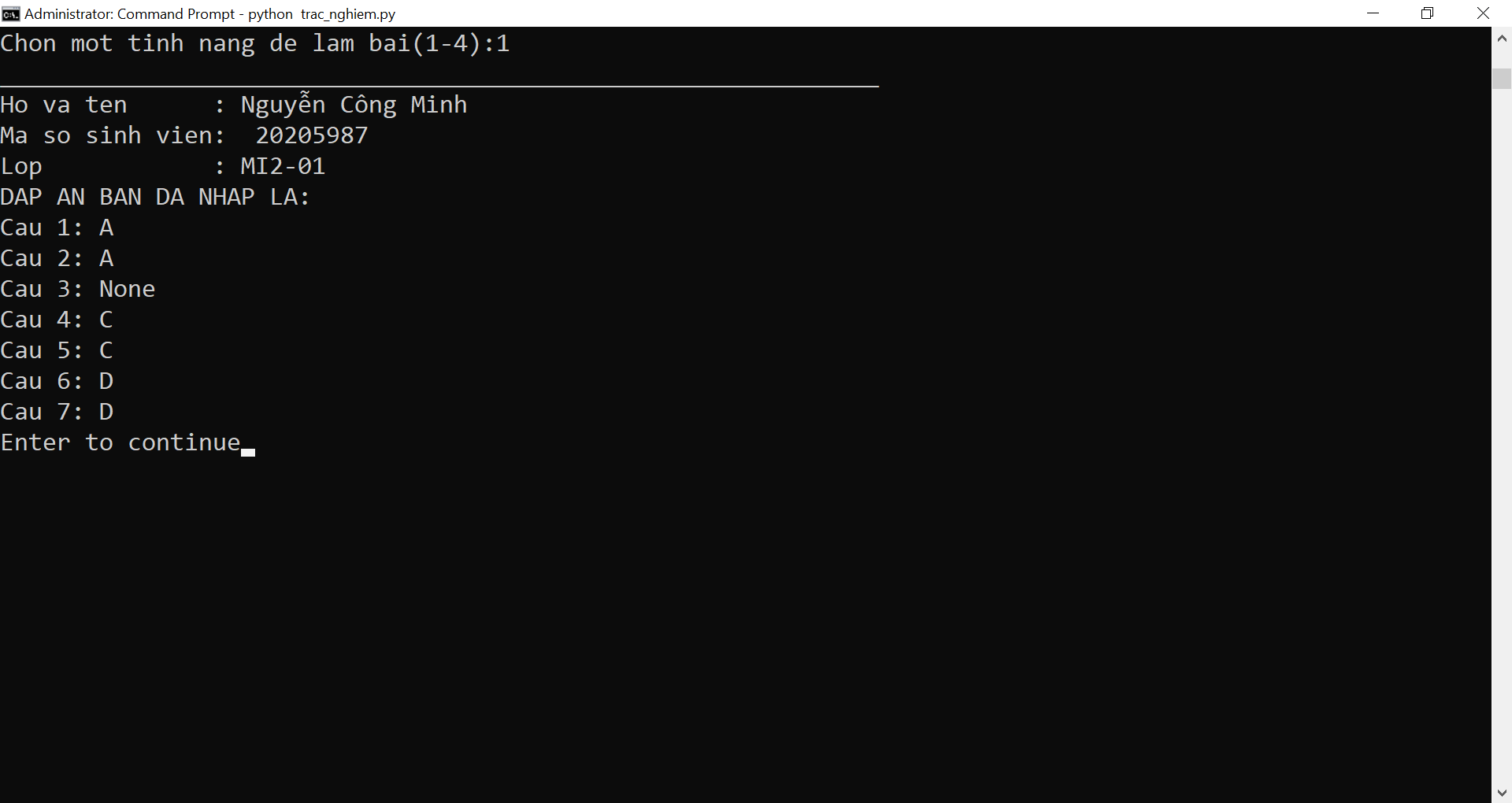


Nhập đáp án trả lời của thí sinh (chấp nhận trường hợp bỏ không điền).

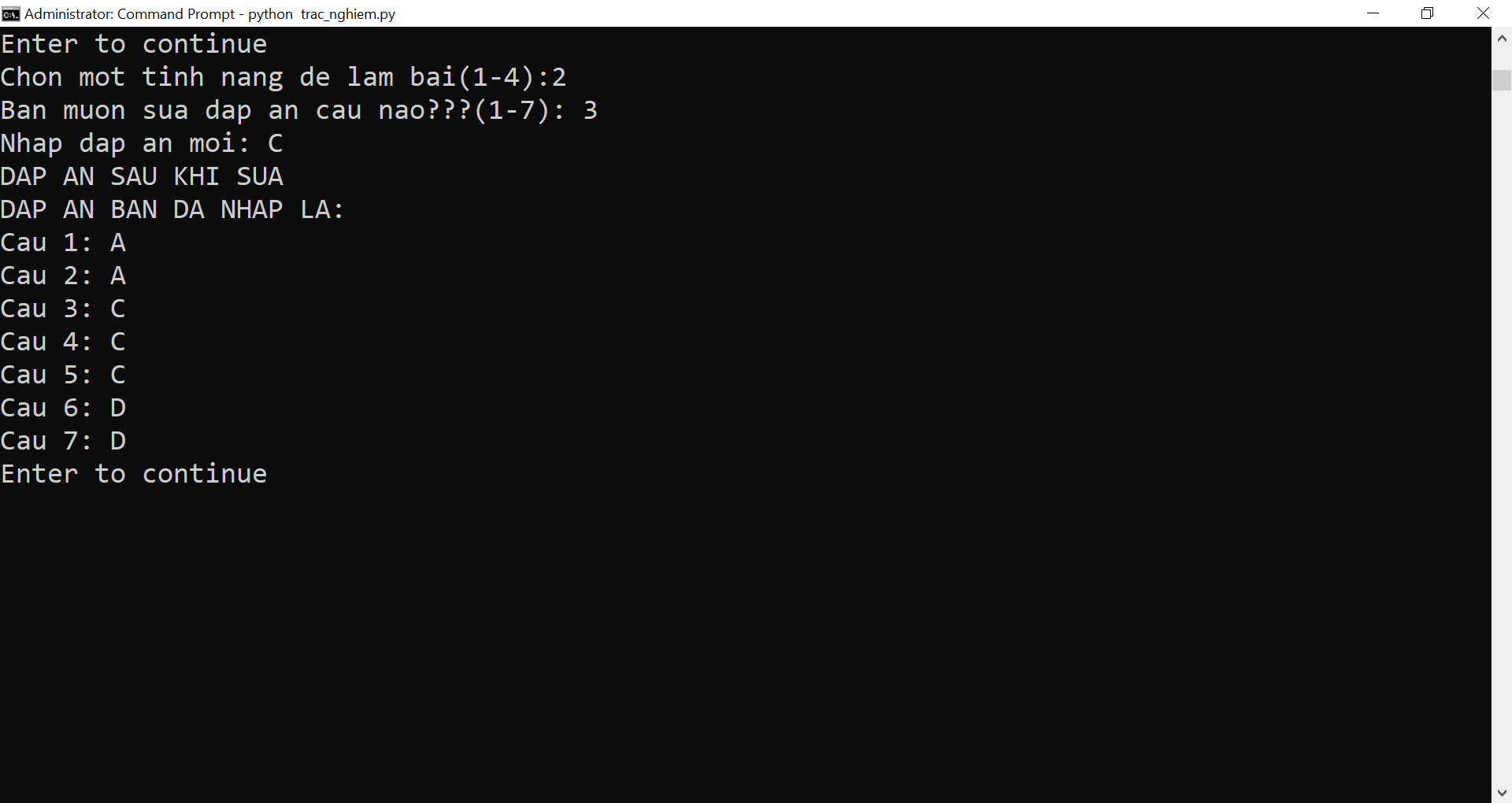
Trường hợp nhập sai đáp án và trường hợp nhập đúng đáp án.



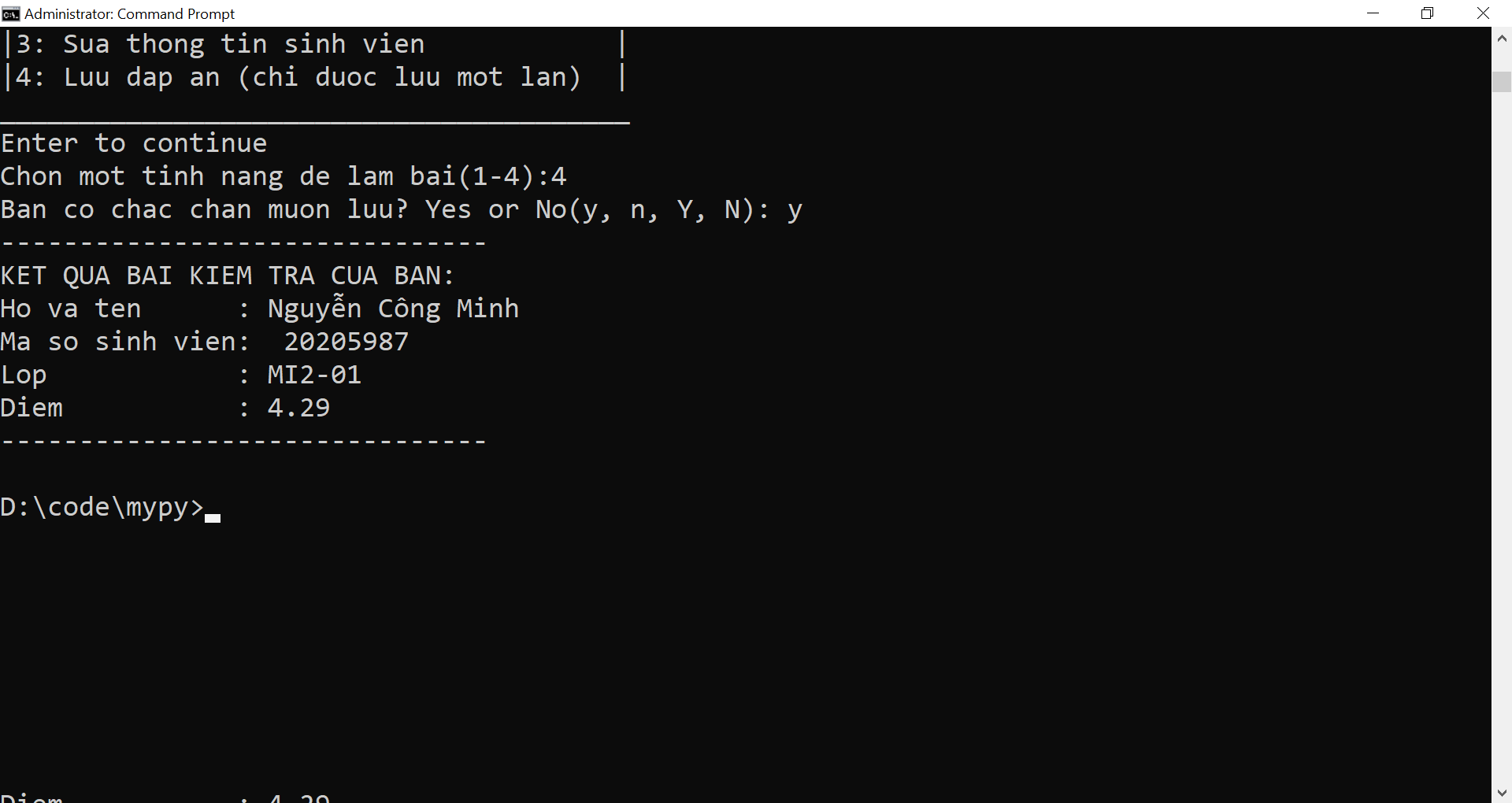
Hiển thị bài làm của thí sinh, đáp án không điền được hiển thị là None.



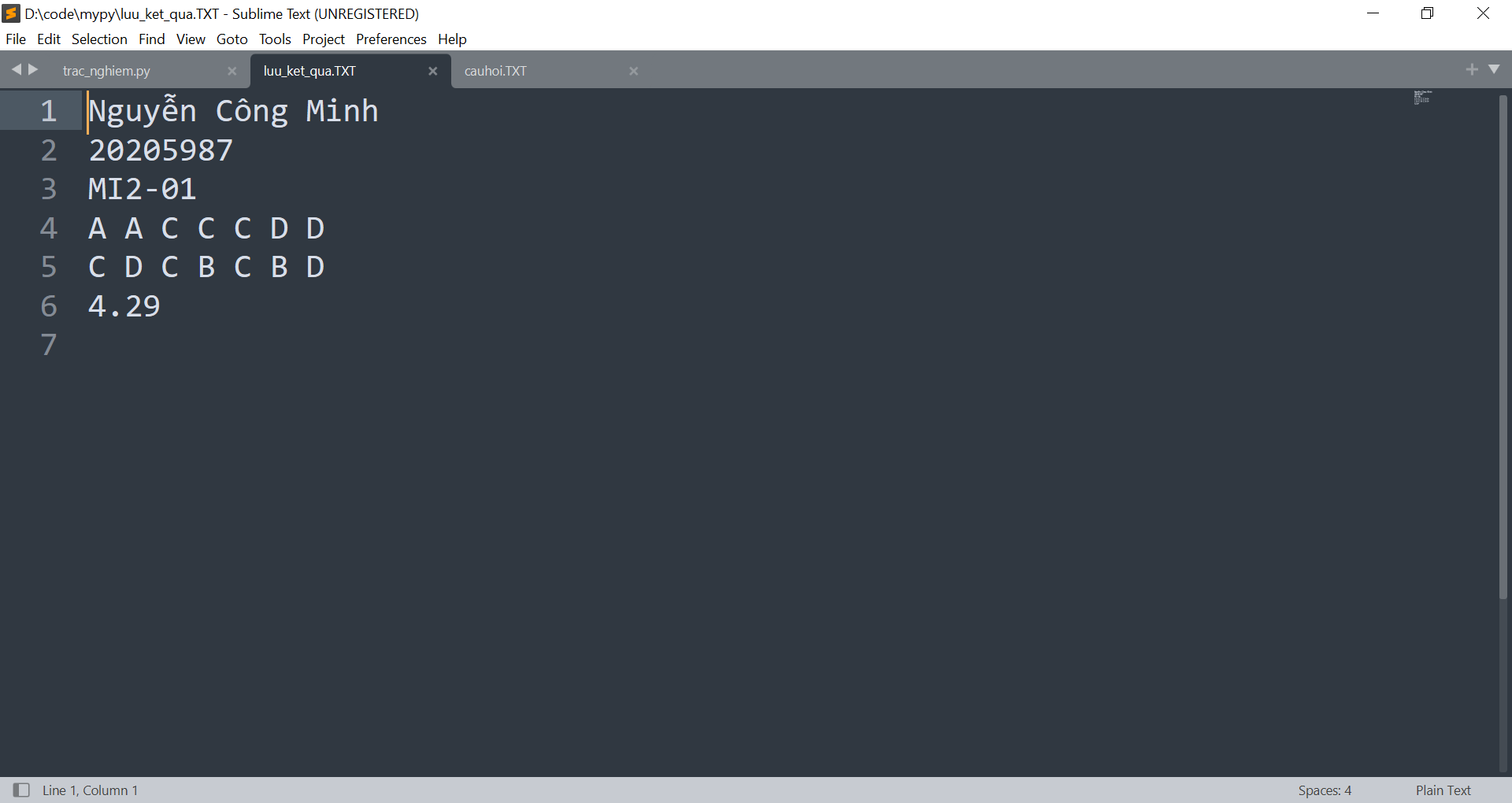
Thực hiện sửa đáp án và hiển thị đáp án sau khi sửa.(Không chấp nhận sửa là không điền).



Lưu thông tin và hiển thị bài làm thí sinh.



Chụp ảnh màn hình file.



# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Slide bài giảng Kỹ thuật lập trình, tác giả TS. Vũ Thành Nam. |