**HỌC VIỆN TÀI CHÍNH**

**BỘ MÔN: LẬP TRÌNH PYTHON CĂN BẢN**

**🙚🙚🕮🙘🙘**



**BÁO CÁO NHIỆM VỤ**

**LỚP 41.2LT**

**VIẾT CHƯƠNG TRÌNH TRA CỨU TỪ ĐIỂN VIỆT-ANH, TRONG ĐÓ FILE TỪ ĐIỂN ĐỂ SỬ DỤNG ĐỊNH DẠNG INI**

**MỤC LỤC**

[**I.** **Thông tin sản phẩm nộp: Nhom\_06.zip** 3](#_Toc146059915)

[**1.** **File chương trình tra cứu từ điển: dictionary\_VIEtoENG\_program.py** 3](#_Toc146059916)

[**2.** **File chuyển đổi dữ liệu từ txt sang ini: convert\_to\_INI.py** 3](#_Toc146059917)

[**3.** **File word thuyết minh nhiệm vụ: Nhom\_06.docx** 3](#_Toc146059918)

[**4.** **Link tải từ điển gốc.txt** 3](#_Toc146059919)

[**II.** **Bảng phân công nhiệm vụ** 4](#_Toc146059920)

[**III.** **Chuyển file .txt sang file .ini** 4](#_Toc146059921)

[**1.** **File .ini** 4](#_Toc146059922)

[**a.** **Khái niệm:** 4](#_Toc146059923)

[**b.** **Quy tắc:** 4](#_Toc146059924)

[**c.** **Ưu điểm** 5](#_Toc146059925)

[**d.** **Nhược điểm** 5](#_Toc146059926)

[**2.** **Chuyển giá trị từ .txt sang .ini** 6](#_Toc146059927)

[**a.** **Nguyên tắc chuyển dữ liệu** 6](#_Toc146059928)

[**b.** **Chi tiết code Python convert file “TuDien\_VietAnh23460.txt”** 7](#_Toc146059929)

[**IV.** **Mô tả code Python file tra cứu từ điển** 10](#_Toc146059930)

[**1.** **Lưu nhật kí ( def write\_log(Mode,Content) )** 10](#_Toc146059931)

[**2.** **Bảng MENU lựa chọn** 10](#_Toc146059932)

[**3.** **Tra từ ( def search\_word() )** 11](#_Toc146059933)

[**4.** **Bổ sung từ ( def add\_word() )** 13](#_Toc146059934)

1. **Thông tin sản phẩm nộp: Nhom\_06.zip**
2. **File chương trình tra cứu từ điển: dictionary\_VIEtoENG\_program.py**

Mô tả file: Chương trình Python có các chức năng:

* *Bảng MENU lựa chọn:* Có 3 thao tác gồm “Tra từ”, “Bổ sung từ mới”, “Kết thúc chương trình”, cho phép lựa chọn thao tác cho đến khi lựa chọn Kết thúc chương trình thì chương trình dừng lại, bảng MENU sẽ lặp lại sau mỗi lần kết thúc các thao tác Tra từ hoặc Bổ sung từ mới
* *Lựa chọn 1: Tra từ:* Nhập từ tiếng Việt cần tra, kết quả tra cứu lấy từ file từ điển định dạng .ini được convert, đưa thông tin của từ ra màn hình, nhật kí tra từ được lưu lại ở file log.txt; cho phép tra cứu lặp lại cho đến khi người dùng ấn Enter thì quay lại bảng MENU lựa chọn.
* *Lựa chọn 2: Bổ sung từ mới:* Thêm từ mới vào từ điển, kiểm tra sự tồn tại của từ nhập vào, nếu từ đã tồn tại thì quay trở lại bảng MENU, nhật kí thêm từ được lưu lại ở file log.txt, cho phép lặp lại việc nhập bổ sung từ đến khi người dùng ấn Enter thì quay lại bảng MENU lựa chọn.
* *Lựa chọn 3: Kết thúc chương trình:* Chương trình dừng lại.

1. **File chuyển đổi dữ liệu từ txt sang ini: convert\_to\_INI.py**

* Chuyển dữ liệu từ file TXT sang file INI theo đúng quy ước của file INI
* Kết quả tạo ra file “TuDien\_VietAnh.ini”, từ điển này được sử dụng làm nguồn dữ liệu cho chương trình tra cứu từ điển.

1. **File word thuyết minh nhiệm vụ: Nhom\_06.docx**
2. **Link tải từ điển gốc.txt**
3. **Bảng phân công nhiệm vụ**

|  |  |
| --- | --- |
| **Thành viên** | **Nhiệm vụ** |
| 1. Thái Thị Hải Yến | * Làm phần: Menu lựa chọn + Nhật kí thao tác |
| 1. Phan Hà An | * Làm phần: Menu lựa chọn + Nhật kí thao tác |
| 1. Nguyễn Thị Kim Anh (Nhóm trưởng) | * Phân chia công việc * Tổng hợp code, viết file code hoàn chỉnh và bản word thuyết minh * Làm phần: Bổ sung từ mới, Chuyển định dạng file |
| 1. Tô Thị Vân Anh | * Làm phần: Chuyển định dạng file |
| 1. Lê Trần Bách | * Làm phần: Tra từ |
| 1. Đinh Thị Thảo Chi | * Làm phần: Tra từ |

1. **Chuyển file .txt sang file .ini**
2. **File .ini**
3. **Khái niệm:**

Định dạng file .INI là một định dạng tệp cấu hình văn bản đơn giản được sử dụng phổ biến trong các ứng dụng và hệ thống. Định dạng này thường được sử dụng để lưu trữ thông tin cấu hình cho các ứng dụng, ví dụ như cài đặt của chương trình, tùy chọn và giá trị khác liên quan.

1. **Quy tắc:**

File .INI bao gồm các phần tử gọi là "sections" (phần) và "keys" (khóa). Mỗi section bao gồm một hoặc nhiều key với giá trị tương ứng. Cú pháp của file .INI bao gồm section và các cặp “key:value” (cặp khóa-giá trị) tương ứng.

Key phải có tên và giá trị tương ứng, được phân tách bằng dấu “=”. Section và key không phân biệt chữ hoa, chữ thường.

Một file có thể bao gồm nhiều section, mỗi section có thể có một hoặc nhiều cặp key:value.

Ví dụ: 1 file “vi\_du.ini” lưu trữ thông tin dưới dạng:

[section1]

key1:value1

key2:value2

[section2]

key3:value3

…

1. **Ưu điểm**

* *Dễ đọc và viết:* File INI sử dụng cú pháp đơn giản, dễ đọc và viết cho con người. Các thông tin cấu hình được lưu trữ dưới dạng cặp key-value, giúp người dùng dễ dàng thay đổi các giá trị cấu hình.
* *Tương thích đa nền tảng:* Định dạng INI đã tồn tại trong một thời gian dài và được hỗ trợ rộng rãi trong nhiều ngôn ngữ lập trình. Điều này có nghĩa là các tệp INI có thể đọc và sửa đổi trên nhiều nền tảng khác nhau.
* *Không cần phụ thuộc vào thư viện bên ngoài:* Để đọc hoặc viết tệp INI, không cần sử dụng thư viện bên ngoài phức tạp. Điều này làm cho việc làm việc với tệp INI trở nên đơn giản và tiết kiệm thời gian.

1. **Nhược điểm**

* *Hạn chế cho các cấu hình phức tạp:* File INI thường không thích hợp cho các ứng dụng hoặc hệ thống có cấu hình phức tạp. Khi cấu hình trở nên rất lớn và phức tạp, việc duy trì và quản lý các tệp INI có thể trở nên khó khăn.
* *Khả năng tổ chức hạn chế:* File INI không cung cấp cơ chế tổ chức dữ liệu mạnh mẽ. Nó chỉ hỗ trợ một cấp độ key-value, nên không thể sắp xếp các thông tin cấu hình thành các nhóm hoặc phân loại chúng một cách hiệu quả.
* *Không hỗ trợ các kiểu dữ liệu phức tạp:* File INI thường chỉ hỗ trợ các kiểu dữ liệu cơ bản như chuỗi, số nguyên và số thực. Nếu bạn cần lưu trữ các kiểu dữ liệu phức tạp như danh sách, đối tượng hoặc mảng, bạn cần phải tự định nghĩa cách biểu diễn chúng trong tệp INI.
* *Khó khăn trong việc bảo mật:* File INI không cung cấp cơ chế bảo mật mạnh mẽ cho thông tin cấu hình. Nó thường được lưu trữ dưới dạng văn bản thuần túy, do đó, nếu cần bảo mật dữ liệu nhạy cảm, bạn cần phải áp dụng các biện pháp bảo mật bổ sung.

1. **Chuyển giá trị từ .txt sang .ini**
2. **Nguyên tắc chuyển dữ liệu**

* File từ điển “TuDien\_VietAnh23460.txt” có dạng:

@Từ tiếng Việt 1

\* Loại từ

-Nghĩa tiếng Anh

=Ví dụ tiếng Việt+Ví dụ chuyển sang nghĩa tiếng Anh

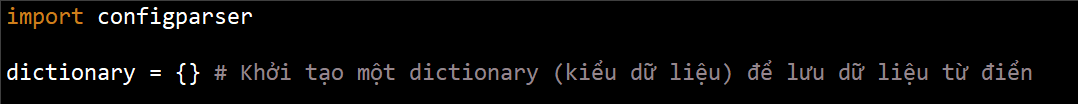
@Từ tiếng Việt 2

…

* Một từ tiếng Việt có thể không có hoặc có nhiều loại từ, nghĩa và ví dụ.
* Lấy từ tiếng Việt làm section, mỗi section có 1 key cố định tên là “definitions”, key này lưu trữ value là toàn bộ giá trị bao gồm loại từ, nghĩa, ví dụ của từ tiếng Việt tương ứng, dấu hiệu phân chia các section là “@”.

1. **Chi tiết code Python convert file “TuDien\_VietAnh23460.txt”**

* Import thư viện “configparser”: thư viện hỗ trợ thao tác với tệp .ini. Khởi tạo biến dictionary={} để lưu dữ liệu dạng từ điển sau khi thao tác với file .txt



* Do trong file “TuDien\_VietAnh23460.txt” có kí tự “%”: ký tự “%” được sử dụng trong Python để thực hiện định dạng chuỗi (string formatting) bằng cách sử dụng variable interpolation, nếu không xử lý kí tự này, chương trình sẽ báo lỗi “ValueError: invalid interpolation syntax”, nên ta thực hiện mở file từ điển để đọc và ghi, dùng hàm replace() thay thế kí tự “%” thành “%%”, ghi đè vào file .txt và đóng file.

A computer code on a black background

Description automatically generated

* Mở file “TuDien\_VietAnh23460.txt” để thực hiện đọc, phân tách các đoạn dựa vào dấu hiệu “@”:

A computer screen with many colorful text

Description automatically generated with medium confidence

+ Dùng vòng lặp for để kiểm tra lần lượt từng dòng.

+ Nếu dòng bắt đầu bằng “@” thì lưu dòng đó vào biến current\_word, nếu dòng không bắt đầu bằng “@” thì lưu vào biến current\_definitions.

+ Sau khi lưu giá trị vào biến current, lưu giá trị của biến current\_definitions vào thành 1 từ điển con có dạng {“definitions” : “current\_definitions”}, rồi lưu current\_word và từ điển con vào biến dictionary nằm ngoài vòng for, sau đó reset giá trị của các biến current về rỗng rồi tiếp tục vòng lặp để lưu giá trị của từ tiếp theo. (Giả sử có current\_word là “a”, current\_definitions là “b” thì từ “a” sẽ được lưu vào biến dictionary, dictionary có dạng:

{

“a” : {“definitions” : “b”}

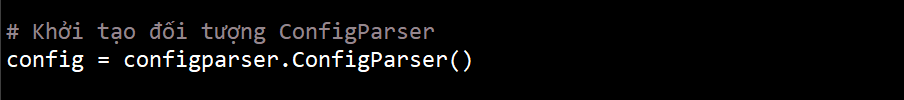
…

}

Các từ tiếp theo được lưu tiếp tương tự vào biến dictionary)

+ Logic của chương trình là khi tìm được dòng bắt đầu bằng “@” thì sẽ lưu dữ liệu từ vừa mới tách được bên trên, nên khi kết thúc vòng lặp, tức là đến cuối file thì ta phải lưu thêm từ cuối vì không còn dòng nào bắt đầu bằng “@” phía sau để chương trình lưu, do đó dùng tiếp lệnh if để lưu từ cuối.

* Khởi tạo một đối tượng ConfigParser từ thư viện configparser (ConfigParser là một class trong thư viện configparser, cung cấp các phương thức để làm việc với file .ini như đọc, ghi, …)



* Dùng for lặp qua các giá trị của biến dictionary, với mỗi giá trị thì tạo một section của từ, key “definitions” với value là giá trị tương ứng của “definitions”, lưu vào đối tượng cấu hình “config”

A black screen with white text

Description automatically generated

* Mở file “TuDien\_VietAnh.ini” với chế độ ghi đè, ghi lại dữ liệu vào file và đóng file.

A computer screen shot of white text

Description automatically generated

* File .txt đã được convert thành công sang file .ini.

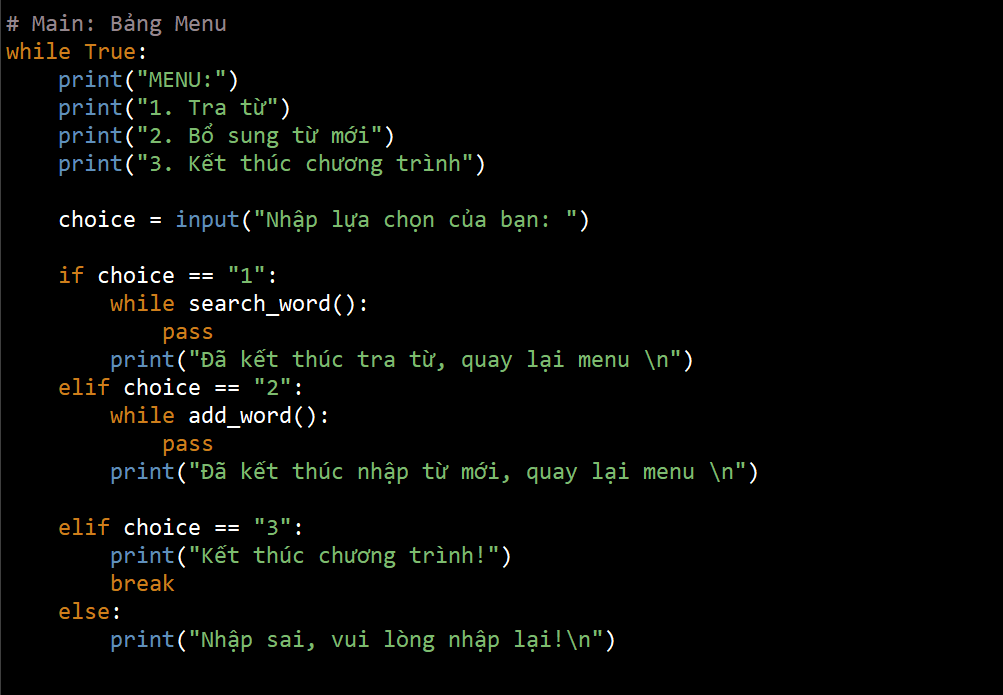
1. **Mô tả code Python file tra cứu từ điển**
2. **Lưu nhật kí ( def write\_log(Mode,Content) )**

**A computer code on a black background

Description automatically generated**

* Mở file “log.txt” để thao tác với mode và nội dung tùy chỉnh
* Biến timestamp lưu lại thời gian thực ngay khi lệnh được chạy, phương thức strftime định nghĩa lại cách hiển thị thời gian: ngày-tháng năm giờ:phút:giây
* Ghi lại nội dung tùy chỉnh (Content) và thời gian vào file “log.txt”
* Đóng file

1. **Bảng MENU lựa chọn**



* Dùng vòng lặp while True để lặp lại số lần không xác định cho bảng MENU
* Đưa ra các lựa chọn 1 2 3, nhập vào lựa chọn, dùng if elif else để rẽ nhánh chương trình:
* Nếu lựa chọn 1: Chạy hàm search\_word(), dùng while lặp lại số lần không xác định
* Nếu lựa chọn 2: Chạy hàm add\_word, dùng while lặp lại số lần không xác định
* Nếu lựa chọn 3: In ra thông báo kết thúc chương trình và break để dừng lại các tiến trình sau, chương trình dừng
* Nếu lựa chọn khác 1,2,3: In ra thông báo nhập sai, quay lại bảng MENU để nhập lại

1. **Tra từ ( def search\_word() )**

**A computer screen with white and green text

Description automatically generated**

**A computer screen with green and white text

Description automatically generated**

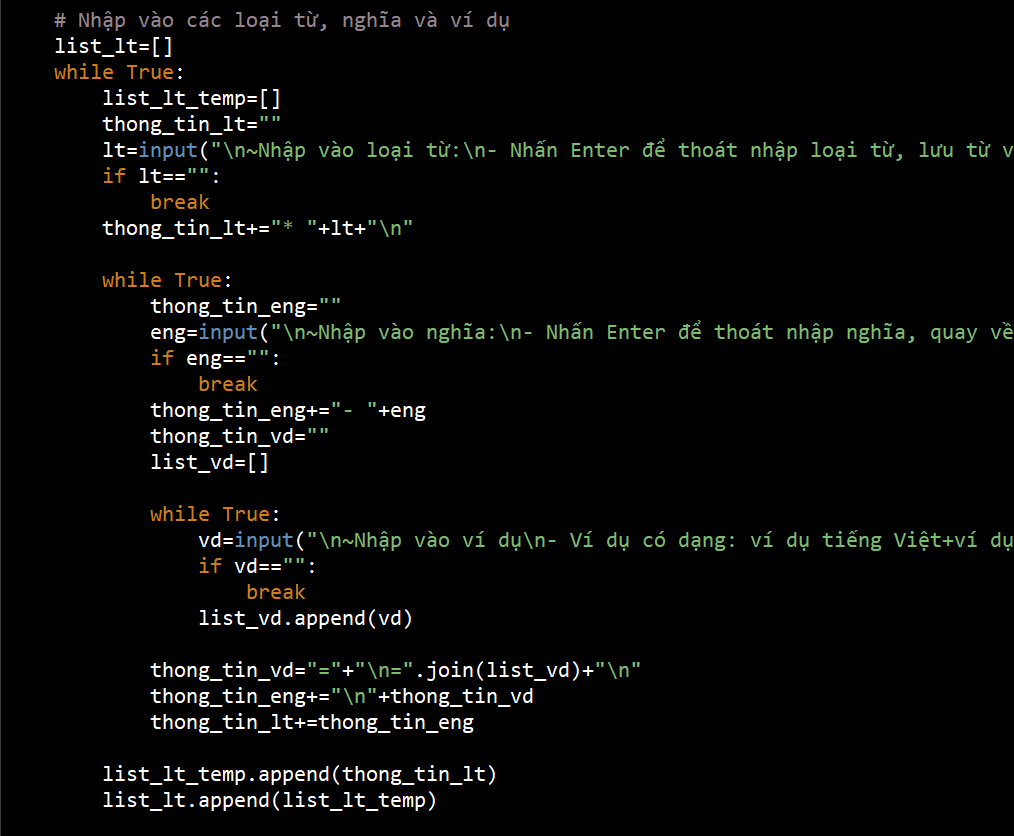
* Nhập vào từ Tiếng Việt cần tra, lưu vào biến tu\_tv\_tra
* Chỉnh sửa lại giá trị của tu\_tv\_tra, loại bỏ khoảng trống ở 2 đầu và biến chuỗi thành chữ thường
* Kiểm tra nếu từ nhập vào là 1 chuỗi rỗng “” (xảy ra khi ấn Enter) thì trả về kết quả False cho vòng lặp while search\_word() để dừng vòng lặp, quay về bẳng MENU lựa chọn.
* Tạo một đối tượng config1 lấy ra class ConfigParser() từ thư viện configparser để đọc file “TuDien\_VietAnh.ini”
* Kiểm tra nếu từ nhập vào là 1 section nào đó ở trong từ điển (từ có trong từ điển) thì in ra thông tin của từ:
* Từ tiếng việt được tìm thấy là 1 section, thông tin của từ là giá trị value của key “definitions”
* Dùng vòng for lặp qua các bộ key:value của section đó, với mỗi key lưu vào biến key, mỗi value tương ứng lưu vào biến value. In ra giá trị value của section đó sau khi đã chuyển đổi kí tự “\* ” thành “ ~Loại từ: ”, kí tự “=” thành “Ví dụ: ”, kí tự “+” thành “\n\tEx: ”
* Lưu vào file log.txt nội dung lich\_su (Content) với mode= “a” (Ghi thêm vào cuối file)
* Nếu từ nhập vào không trùng với bất kì section nào trong file từ điển thì in ra kết quả là không tìm thấy từ trong từ điển và lưu vào file log nội dung lich\_su1 (Content) với mode “a”
* Trả về kết quả True cho vòng lặp while search\_word(), vòng lặp tiếp tục (Việc tra từ được lặp lại vô số lần cho đến khi ấn Enter để trả về kết quả False, dừng vòng lặp)

1. **Bổ sung từ ( def add\_word() )**

**A computer screen with text

Description automatically generated**

* Nhập vào từ Tiếng Việt và kiểm tra nếu từ đã có trong từ điển thì lưu nội dung lich\_su2 vào file log.txt, return False để thoát thao tác bổ sung từ mới, quay về MENU (nguyên tắc giống phần tra từ)
* Tạo biến thong\_tin\_end= “” để lưu thông tin của từ ở dạng str, biến thong\_tin\_list=[] để lưu thông tin của từ ở dạng list



* Tạo biến list\_lt=[] nằm ngoài vòng while (1)
* Tạo vòng while (1), tạo 1 biến list\_lt\_temp=[] để lưu thong\_tin\_lt ở dạng list, biến thong\_tin\_lt= “” để lưu loại từ (loại từ 1) nhập vào ở dạng str. Lúc này biến thong\_tin\_lt có dạng: “\* Loại từ 1”
* Sau đó tạo vòng while (2), biến thong\_tin\_eng= “” để lưu nghĩa tanh nhập vào ở dạng str (nghĩa tanh 1), lúc này biến thong\_tin\_eng có dạng “- Nghĩa tanh 1”, sau đó đặt thong\_tin\_vd= “” và list\_vd=[] ở ngoài vòng while (3), (list\_vd để lưu ví dụ nhập vào (dạng str) và append vào list, còn thong\_tin\_vd lưu ví dụ của list\_vd sau khi đã join các phần tử lại (dạng str))
* Sau đó tạo vòng while (3), vs các vd nhập vào thì append vào list\_vd, cho đến khi ấn Enter (thoát vòng while (3)), trước khi quay lại thực hiện vòng while (2) thì thực hiện tiếp các lệnh:

+ gán giá trị sau khi đã join các phần tử của list\_vd vào thong\_tin\_vd (ở dạng str). lúc này biến list\_vd có dạng: [“vd 1+ex 1”, “vd 2+ex 2”, “vd n+ex n”], biến thong\_tin\_vd có dạng:

        “=vd 1+ex 1

    =vd 2+ ex 2

     =vd n+ex n”

+ gán giá trị của biến thong\_tin\_eng là nghĩa nhập vào (1 nghĩa) + thong\_tin\_vd (đây là các ví dụ tương ứng của nghĩa). Lúc này biến thong\_tin\_eng có dạng:

    “- Nghĩa tanh 1

   =vd 1+ex 1

  =vd 2+ex 2

     =vd n+ex n”

+ gán giá trị của biến thong\_tin\_lt là loại từ + thong\_tin\_eng. Lúc này biến thong\_tin\_lt có dạng:

     “\* Loại từ 1

     - Nghĩa tanh 1

      =vd 1+ex 1

     =vd 2+ex 2

     =vd n+ex n”

* Sau đó quay trở lại thực hiện vòng while (2), biến thong\_tin\_eng được reset về rỗng để tiếp tục lưu thông tin nghĩa 2 và ví dụ của nghĩa 2, ...cho đến khi nhập đủ các nghĩa của loại từ 1, cho đến khi ấn Enter để xác nhận việc dừng nhập nghĩa thì các câu lệnh sau được thực hiện:

+ append giá trị của thong\_tin\_lt vào list\_lt\_temp, lúc này list\_lt\_temp có dạng:

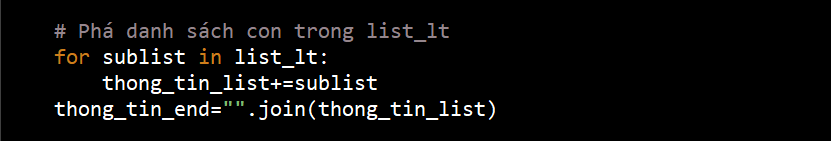
[“\* Loại từ 1\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n”]

+ append giá trị của list\_lt nằm ngoài vòng while (1), lúc này list\_lt gồm phần tử là các danh sách con, mỗi danh sách con gồm 1 phần tử là thong\_tin\_lt:

[[“\* Loại từ 1\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n”]]

* Sau đó vòng lặp while (1) tiếp tục đc thực hiện, biến list\_lt\_temp dc reset về list rỗng để lưu thong\_tin\_lt (thông tin về loại từ 2, nghĩa và ví dụ của loại từ 2), vòng lặp while (2) được thực hiện, biến thong\_tin\_eng đc reset về chuỗi rỗng để lưu thông tin nghĩa và thong\_tin\_vd (ví dụ của nghĩa tương ứng), cho đến khi nhập đủ loại từ của từ nhập vào và ấn Enter, lúc này biến list\_lt lưu toàn bộ thông tin về từ tv cần bổ sung ở dạng:

[[“\* Loại từ 1\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n”],...,[“\* Loại từ n\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n”]]



* Dùng vòng for phá các danh sách con ở trong list\_lt, mỗi danh sách con được phá ra, lưu vào biến thong\_tin\_list (có thể hiểu là list\_lt đang ở dạng danh sách gồm các phần tử là danh sách con, bây giờ cần 1 biến (là biến thong\_tin\_list) lưu lại thông tin của list\_lt ở dạng danh sách gồm các phần tử là chuỗi kí tự (các phần tử này là phần tử của danh sách con nằm trong list\_lt)). Sau khi chạy for, biến thong\_tin\_list có dạng:

[“\* Loại từ 1\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n” , ... , “\* Loại từ n\n- Nghĩa tanh 1\n=vd 1+ex 1\n=vd 2+ex 2\n=vd n+ex n”]

* Gán giá trị cho biến thong\_tin\_end là thong\_tin\_list sau khi đã join lại thành chuỗi kí tự, biến thong\_tin\_end có dạng: (đây chính là thông tin cuối cùng của từ Tiếng Việt nhập vào để lưu vào từ điển)

“\* Loại từ 1

- Nghĩa tanh 1

=vd 1+ex 1

=vd 2+ex 2

=vd n+ex n

...

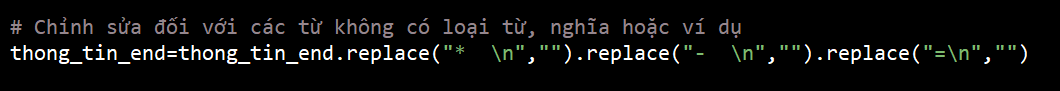
\* Loại từ n

- Nghĩa tanh 1

=vd 1+ex 1

=vd 2+ex 2

=vd n+ex n”



* Ở trên thao tác có nói, nếu từ không có loại từ và nghĩa thì ấn Space, hết ví dụ thì ấn Enter, do đó sẽ tạo ra các kí tự “\* ” (dấu \* và 2 dấu cách), “- ” (dấu – và 2 dấu cách), “=” (dấu = và kí tự enter tức là chuỗi rỗng), nghĩa là nếu có những kí tự này thì sẽ loại bỏ khỏi thông tin của từ => replace nó thành chuỗi rỗng “”

A computer screen with white and green text

Description automatically generated

* Sau đó lưu thành dạng từ điển: tu\_tv vào section, thong\_tin\_end vào value của key “definitions”, rồi ghi vào từ điển “Tudien\_VietAnh.ini” với mode= “a” (ghi thêm vào cuối file)
* Ghi lại vào file log.txt nội dung lich\_su2 với mode= “a” và trả về kết quả True cho vòng lặp while add\_word()