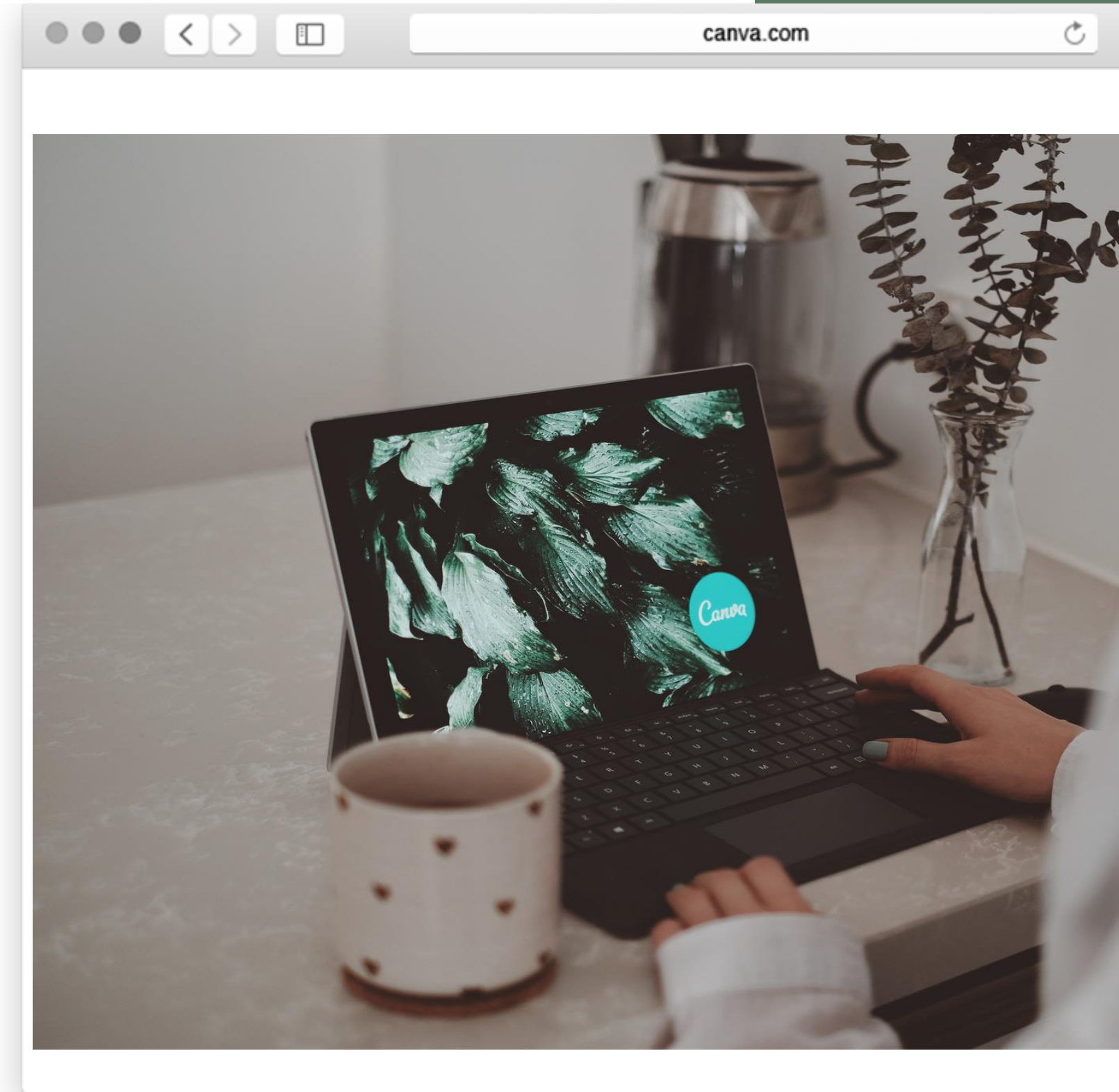


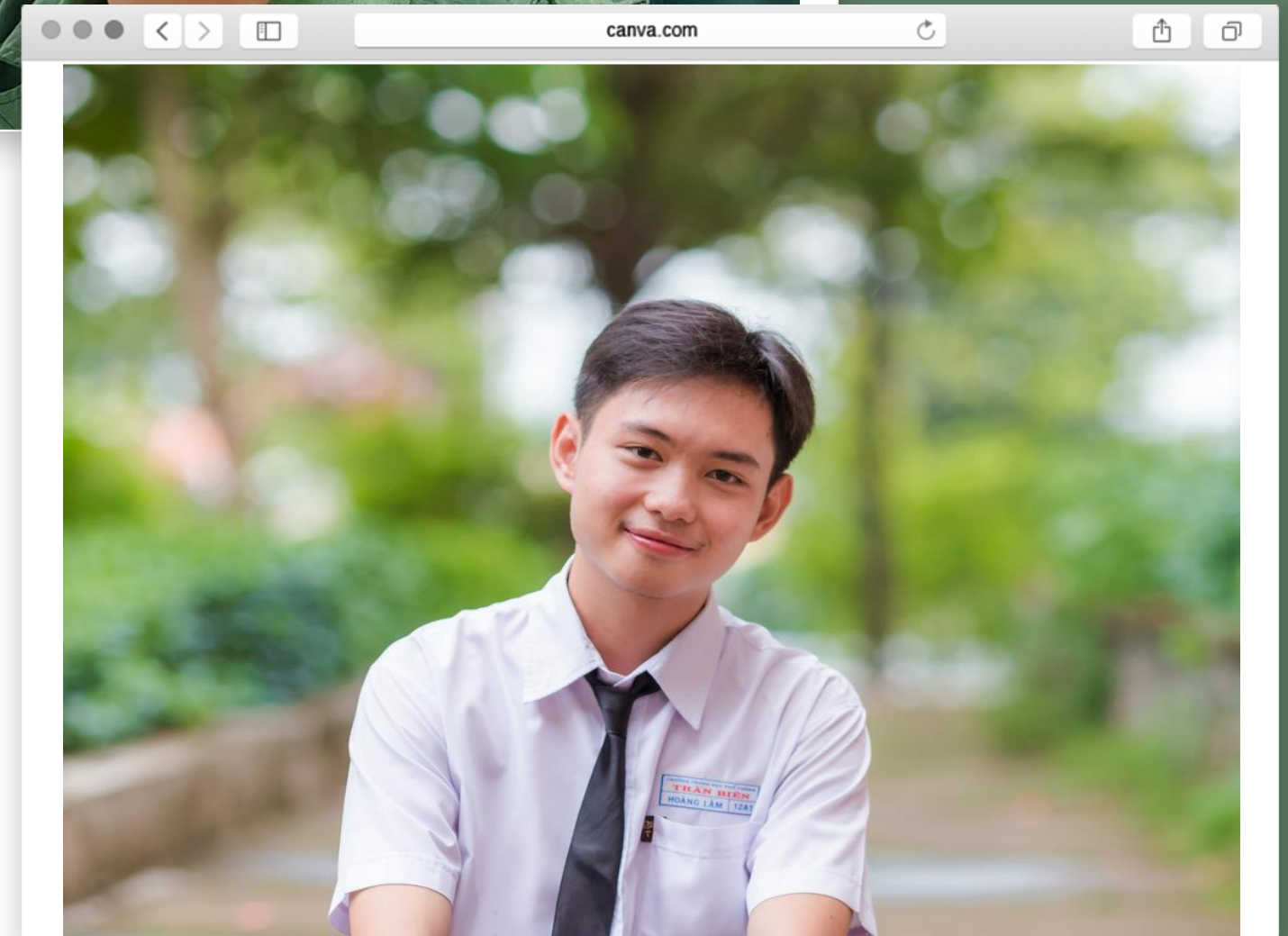
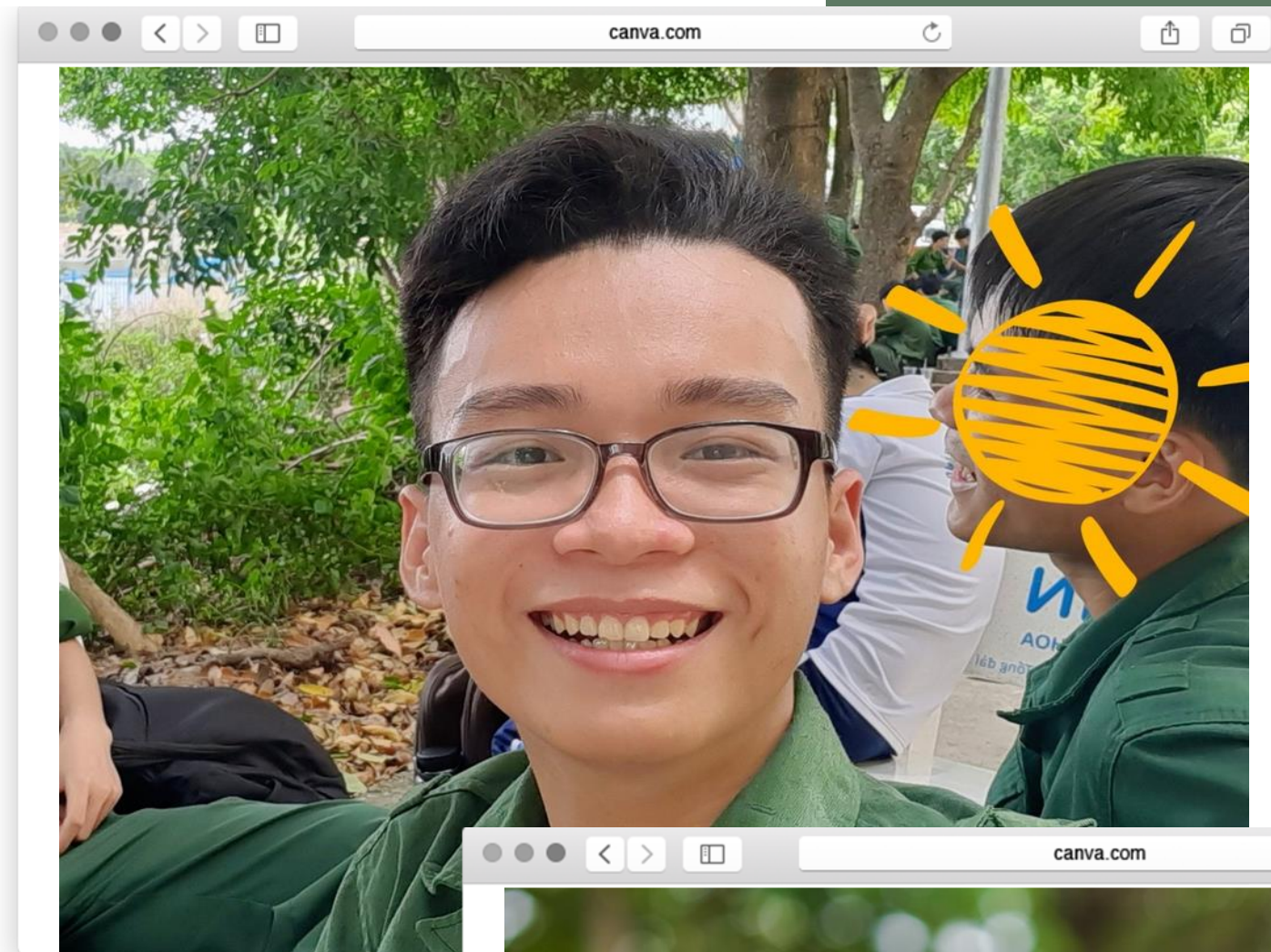
# Search Engine

Đồ án cuối kì  
Kỹ thuật lập trình



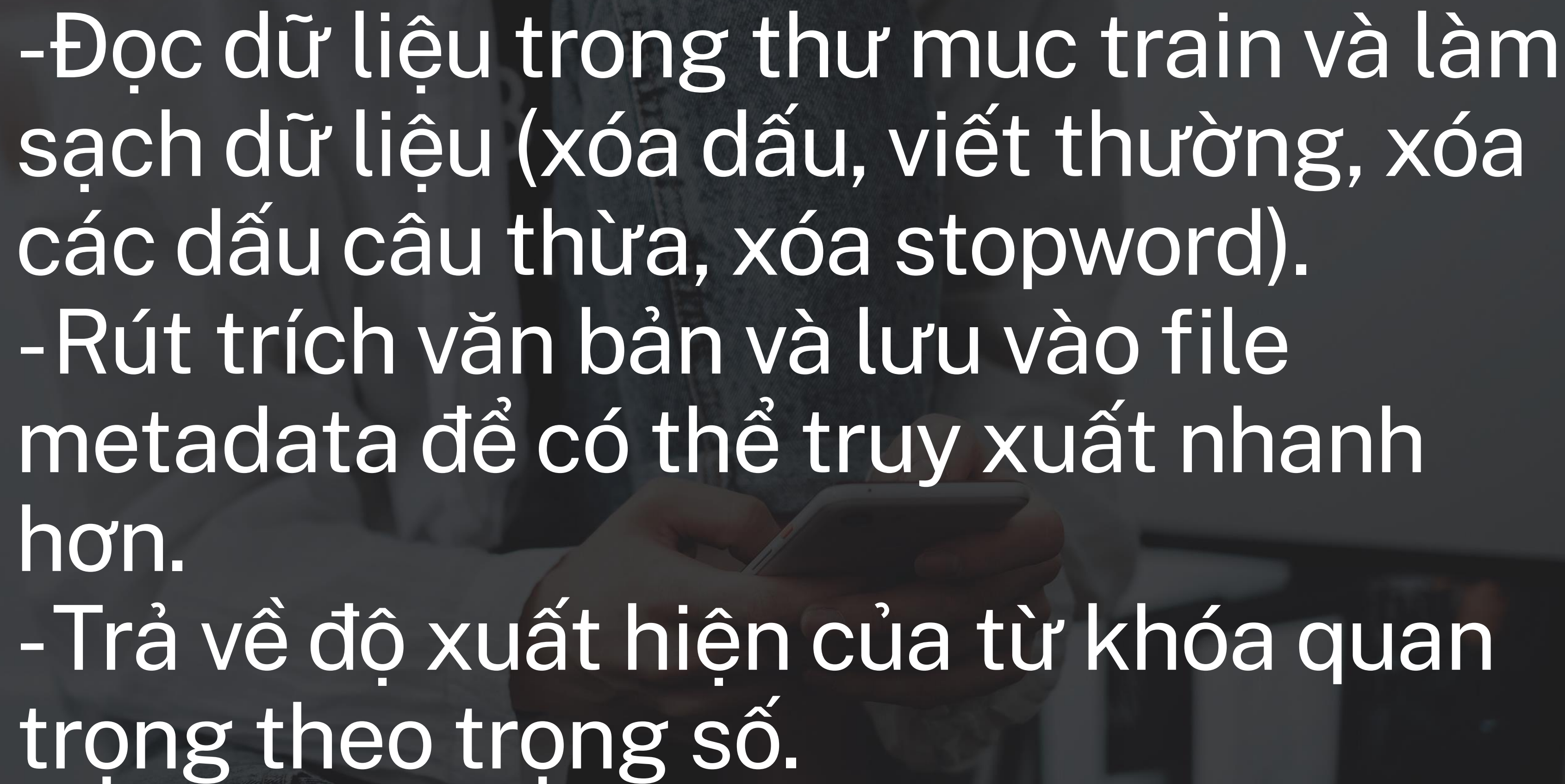
Thành viên:  
Trần Kiều Minh Lâm  
Nguyễn Hoàng Lâm  
Lớp: 20CTT1TN

---





Phân tích đề: để truy  
vấn trong tập dữ liệu lớn  
với tốc độ nhanh, ta cần  
tạo file metadata.

- 
- Đọc dữ liệu trong thư mục train và làm sạch dữ liệu (xóa dấu, viết thường, xóa các dấu câu thừa, xóa stopwords).
  - Rút trích văn bản và lưu vào file metadata để có thể truy xuất nhanh hơn.
  - Trả về độ xuất hiện của từ khóa quan trọng theo trọng số.



## FILE

Lưu các giá trị của 1 file.



## TOPIC

Lưu các giá trị của topic, một topic gồm nhiều file



## DATA

Lưu nhiều topic.

# CÁC STRUCT SỬ DỤNG TRONG ĐỒ ÁN





## FILE

Bao gồm: tên file, đường dẫn file.

isDel(): đánh dấu file đã bị xóa, dùng khi truy vấn xóa file từ người dùng.

+ appear[3]: khi có truy vấn Search, dùng để lưu lại số lần xuất hiện trong file của các từ khóa người dùng nhập vào.

+ grams[3]: lưu lại các từ monogram (grams[0]), digram (grams[1]), trigram (grams[2]).

+ rates[3]: lưu lại phần trăm số lần xuất hiện các từ grams tương ứng.



## TOPIC

Bao gồm: numFiles: số lượng file trong topic,  
cntDel: số lượng file đã đánh dấu xóa (dùng cho  
việc thêm và xóa file).

+ files: là 1 mảng gồm nhiều File.

+ del: giải phóng vùng nhớ.

```
struct Topic {  
    int numFiles = 0;  
    int cntDel = 0;  
    string path;  
    string name;  
    File* files;  
    void del();  
};
```





## DATA

Bao gồm:

+ numTopic: số lượng topic có trong data.

+ path: đường dẫn tới data.

```
struct Data {  
    int numTopic;  
    Topic* topics = nullptr;  
    string path;  
    void del();  
};
```





## VECTORINT

Struct lưu trữ mảng động kiểu int



## VECTORSTR

Struct lưu trữ mảng động kiểu string



## PAIRSI

Lưu một cặp giá trị string, int



## VECTORPSI

Lưu mảng động một cặp giá trị string, int

# CÁC STRUCT SỬ DỤNG TRONG ĐỒ ÁN



## VectorInt và VectorStr

Bao gồm:

- + size: Số lượng phần tử.
- + mảng động int và string.

Các chức năng:

- + pushBack: thêm phần tử, sort: sắp xếp.
- + init: khởi tạo, del: xóa mảng động.
- + lowerBound: (VectorStr) hàm dùng trong tìm kiếm nhị phân, unique: xóa các phần tử trùng nhau.





## PairSI và VectorPSI

Bao gồm:

- + int a: lưu giá trị int, string s: lưu giá trị string (PairSI).

- + VectorPSI: Lưu mảng động các PairSI.

Các chức năng:

- + cmpPSI: so sánh 2 PairSI.

- + pushBack: thêm phần tử, sort: sắp xếp.

- + init: khởi tạo, del: xóa mảng động.

- Sử dụng các thư viện: `<locale>`, `<codecvt>`, `<fstream>`.
- Dùng hàm đọc hai định dạng file (UTF16 và UTF8).
- Kiểm tra file là UTF8 hay UTF16 để chọn cách đọc phù hợp.
- File UTF16 sẽ có 3 byte đầu chứa BOM, còn UTF8 thì không chắc.

ĐỌC FILE  
VĂN BẢN  
TIẾNG VIỆT  
UTF8 VÀ  
UTF16



# Tiền xử lí văn bản

Làm cho văn bản tiếng việt  
trở nên đơn giản, dễ xử lí

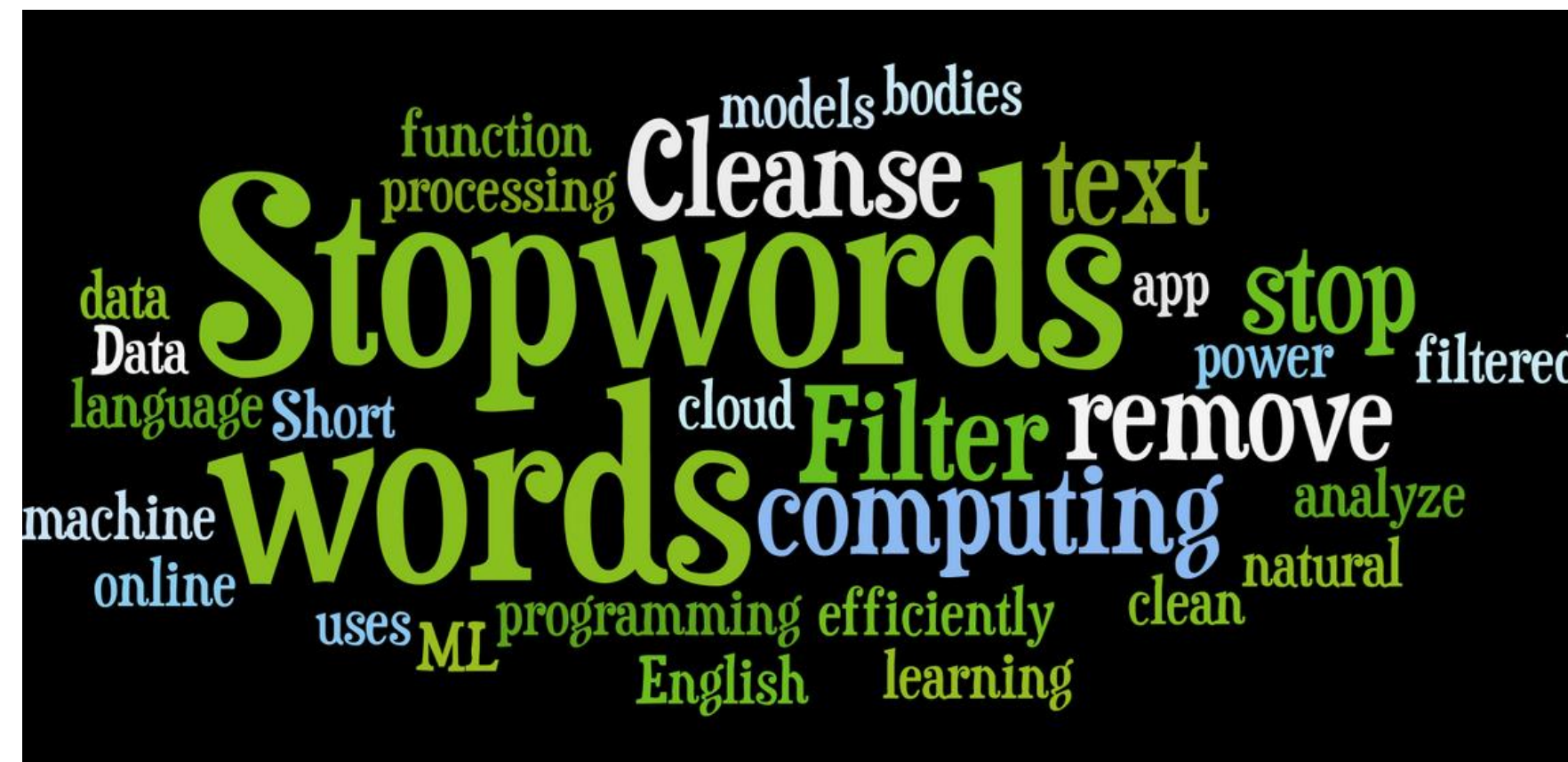


Các bước tiền xử lí:

- Xóa dấu: xóa dấu văn bản, chuyển về lưu dạng wstring.
- lowercase: chuyển tất cả viết in thành thường.
- fixword: xóa các dấu không cần thiết: “.,/;:-\_)(~!@#\$%^&\*.\\n”

# Tiền xử lí văn bản

Làm cho văn bản tiếng việt  
trở nên đơn giản, dễ xử lí



Xóa stopword: Stopword là những từ khóa không quan trọng trong văn bản, xóa đi để tìm kiếm văn bản chính xác hơn.



# Tiền xử lí văn bản

Làm cho văn bản tiếng việt  
trở nên đơn giản, dễ xử lí

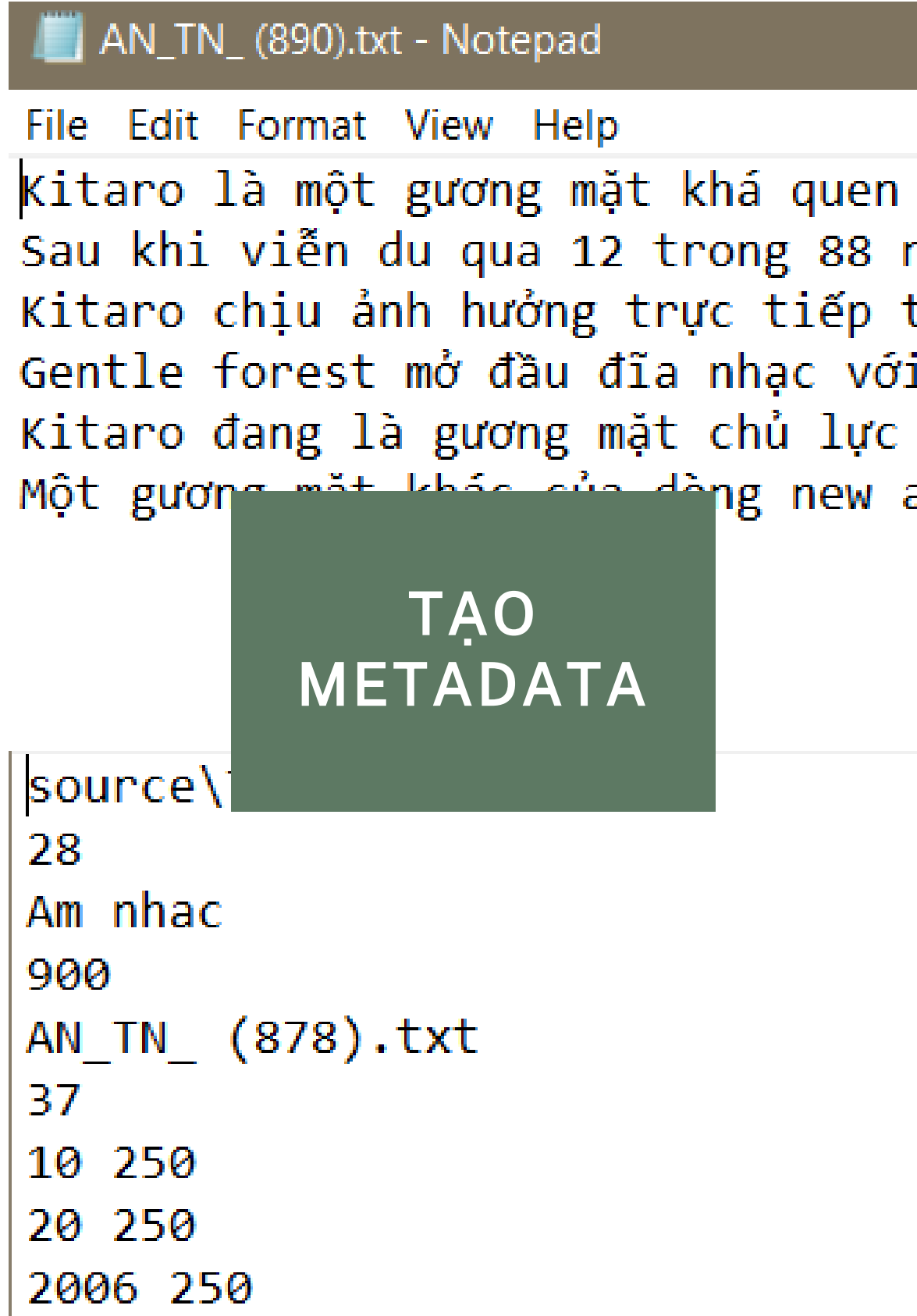
- Đọc file stopwords, tiền xử lí. Tách từng dòng stopwords thành từng token, lưu vào struct VectorStr.
- Sắp xếp các token stopwords tăng dần theo thứ tự từ điển.
- Tiến hành xóa stopwords: duyệt qua các các từ trong văn bản, sử dụng chặt nhị phân để tìm những vị trí xuất hiện của từ trong VectorStr stopwords. So khớp từ trong văn bản với từ trong stopwords, nếu giống nhau thì xóa đi.

## Văn bản gốc

Phức tạp khó xử lý, tìm kiếm lâu

## Văn bản sau khi rút trích vào metadata

Đơn giản, dễ xử lý, tìm kiếm  
nhANH





1

Dùng vòng lặp qua từng file để rút trích dữ liệu từng file một. dùng hàm `extractKeyword`

2

## EXTRACTKEYWORD

Lấy ra những từ xuất hiện nhiều lần trong 1 file. Dùng hàm `countApperance`. Kiểm tra nếu % từ 1~10 thì thêm vào metadata. Nếu văn bản dưới 50 từ thì lấy toàn bộ.

3

## COUNTAPPERANCE

Tính phần trăm từng từ một. cách tính: số lần xuất hiện của từ đó chia cho tổng số từ.

# TẠO FILE METADATA

```
metadata.txt - Notepad
File Edit Format View Help
|source\Train\new train
28
Am nhạc
900
AN_TN_ (878).txt
37
10 250
20 250
2006 250
4 500
am 250
chuong 250
dien 250
gia 250
giang 250
giot 250
ha 250
hcm 250
hen 250
hien 250
hoang 500
huong 250
khuc 250
lac 250
le 250
liveshow 250
luu 250
mau 250
<
```

# File metadata

**Dòng 1:** đường dẫn tới thư mục để train ra metadata.

**Dòng 2:** số lượng topic có trong metadata.

Tiếp theo là nội dung các topic, có định dạng là:

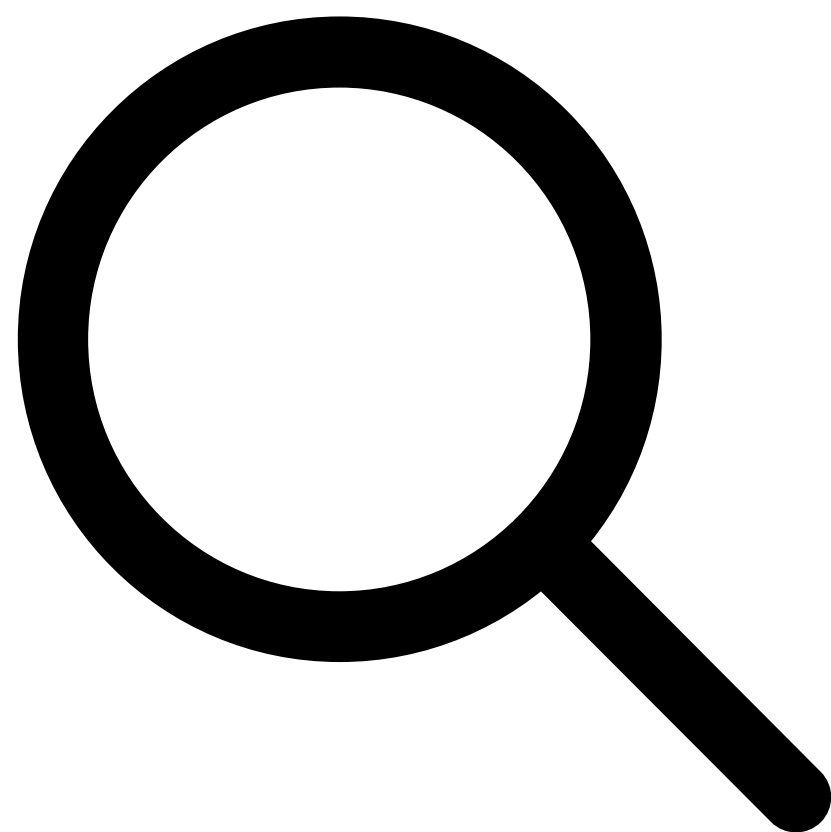
- **Dòng 1:** tên topic
- **Dòng 2:** số lượng file có trong topic đó.

Tiếp theo là nội dung các file trong topic đó, có định dạng là:

- + **Dòng 1:** tên file
- + **Dòng 2:** Số lượng từ trong grams[i], với i = 0, 1, 2.

Các từ của grams[i] trên từng dòng, với i = 0, 1, 2





# SEARCH

- Input người dùng nhập từ bàn phím.
- Tiền xử lý xâu input.
- Duyệt qua input. so sánh với metadata, khớp thì thêm chỉ số rate vào appear.
- Duyệt qua từng File, tính toán điểm của từng file bằng  $\text{sum}(\text{appear}[i] * \text{weight}[i])$ .  
 $\text{weight}[0] = 0,75$  (monogram),  $\text{weight}[1] = 1,25$  (digram),  $\text{weight}[2] = 1,5$  (trigram)
- Sắp xếp và lấy 25 kết quả cao nhất.



## Add File

- Nhập đường dẫn file, kiểm tra file tồn tại.
- Train file và thêm vào topic "other" trong metadata.

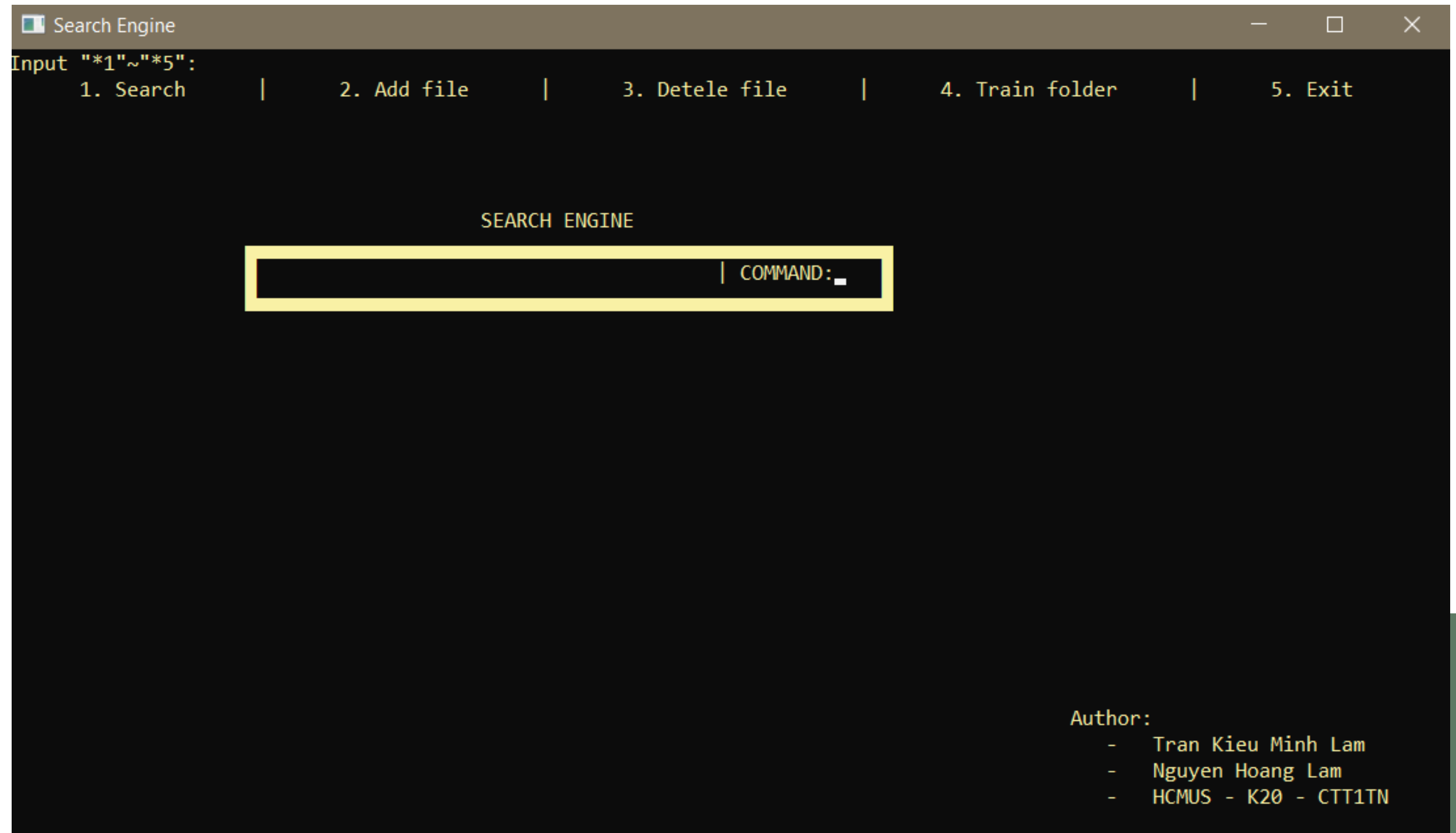


## Delete File

- Nhập tên file cần xóa.
- Kiểm tra file đã có hay chưa. Dùng hàm FindFile() trả về vị trí của file. Đánh dấu isDel = true.



# Giao diện người dùng



# Giao diện người dùng

\*1: Search → Nhập từ khóa tìm kiếm. Sau đó:

-1 → quay trở về menu

0 → chuyển về trang trước.

1-5 → xem chi tiết văn bản 1-5.

6 → chuyển đến trang sau.

\*2: Add file → Nhập vào đường dẫn đến file cần thêm

\*3: Delete file → Nhập vào tên file cần xóa

\*4: Train Folder → Nhập vào đường dẫn đến thư mục cần train.

\*5: Exit → Thoát chương trình, đồng thời lưu lại dữ liệu.



# Thư viện & nguồn tham khảo

-Đọc file UTF8, UTF16

<fcntl.h> <io.h> <locale> <codecvt>

-Thư viện cơ bản

<iostream> <string> <filesystem> <fstream> <Windows.h>

<conio.h>

-Các nguồn tham khảo

<https://stackoverflow.com/>

<https://codelearn.io/>

<https://www.geeksforgeeks.org/>

<https://github.com/>

<https://www.cplusplus.com/reference/>



Thank for watching