**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo with arrows in a white background

Description automatically generated

**ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

**XÂY DỰNG ỨNG DỤNG QUẢN LÍ DANH SÁCH SINH VIÊN**

Người hướng dẫn: **ThS Trần Hồ Thủy Tiên**

Sinh viên thực hiện:

**Mai Thanh Tân LỚP: 23T\_DT4**

**Võ Hoàng Trí LỚP: 23T\_DT4**

**NHÓM: 23Nh13B**

**Đà Nẵng, 05/2024**

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ thông tin phát triển mạnh mẽ hiện nay, việc áp dụng công nghệ vào việc quản lý và xử lý thông tin đã trở thành một yếu tố thiết yếu đối với mọi lĩnh vực. Đặc biệt, trong môi trường giáo dục, tại các trường đại học và cao đẳng, việc quản lý sinh viên đã trở thành một việc cấp thiết, chiếm vai trò vô cùng quan trọng. Hệ thống quản lý truyền thống dựa trên giấy tờ hay các file Excel đơn giản không còn đáp ứng được yêu cầu về tính chính xác, hiệu quả và khả năng truy cập thông tin nhanh chóng.

Sự xuất hiện của ứng dụng quản lí danh sách sinh viên đã mở ra cơ hội mới, giúp cho việc quản lí danh sách sinh viên trở nên dễ dàng và hiệu quả hơn. Việc xây dứng ứng dụng quản lí danh sách sinh viên không chỉ tối ưu hóa quá trình quản lí, giảm thiểu các sai sót có thể xảy ra trong quá trình quản lí mà còn cung cấp các công cụ hỗ trợ giúp các công việc khác như: tra cứu, cập nhật, phân tích dữ liệu… một cách nhanh chóng và chính xác.

Chính vì nhận thức về những điều trên, đồ án “Xây dựng ứng dụng quản lí danh sách sinh viên” được chúng tôi thực hiện nhằm mục đích tạo ra ứng dụng nhằm quản lí thông tin sinh viên hiệu quả, nhanh chóng và tiện lợi, giúp giải quyết những khó khăn và thách thức mà các cơ sở giáo dục đang gặp phải trong công tác quản lí sinh viên. Chúng tôi đã tận dụng những kiến thức và kĩ năng về lập trình để xây dựng một hệ thống linh hoạt, dễ sử dụng, đáp ứng được các yêu cầu đa dạng của người dùng.

Đồ án của chúng tôi gồm một giao diện thân thiện, hướng đến việc dễ sử dụng, truy cập linh hoạt, phù hợp với mọi người dùng giúp việc quản lí sinh viên diễn ra đầy hiệu quả hơn. Hệ thống cho phép quản lí thông tin cá nhân của sinh viên như: Mã số sinh viên, Họ và tên, Ngày/tháng/năm sinh và nhiều thông tin khác, tạo điều kiện thuận lợi cho việc quản lí thông tin của các sinh viên.

Đồ án lập trình “Xây dựng ứng dụng quản lí danh sách sinh viên” không chỉ là một dự án cá nhân mà là sự kết hợp tinh thần đồng đội, sự hỗ trợ và cống hiến hết mình của cả nhóm. Chúng tôi xin gửi lời cảm ơn chân thành đến sự hỗ trợ, giúp đỡ nhiệt tình cũng như sự đóng góp ý kiến quý báu của các giảng viên đã hướng dẫn và hỗ trợ chúng tôi trong quá trình thực hiện dự án này.

**MỤC LỤC**

1. TỔNG QUAN ĐỀ TÀI 1
2. CƠ SỞ LÝ THUYẾT 2
   1. Ý TƯỞNG 2
   2. CƠ SỞ LÝ THUYẾT 2
      1. Kiểu dữ liệu struct 2
      2. Mảng một chiều 2
      3. Mảng một chiều chứa phần tử có kiểu dữ liệu struct 2
      4. Con trỏ 3
      5. Chuỗi và các hàm xử lý chuỗi 4
      6. File và các hàm thao tác file 5
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 5
   1. PHÁT BIỂU BÀI TOÁN 5
   2. CẤU TRÚC DỮ LIỆU 5
   3. THUẬT TOÁN 6
      1. Thuật toán sắp xếp sinh viên 6
      2. Thuật toán thêm sinh viên 7
      3. Thuật toán cấp mã sinh viên 8
      4. Thuật toán cấp email sinh viên 8
      5. Thuật toán tìm kiếm sinh viên 8
      6. Thuật toán xóa sinh viên 8
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 9
   1. TỔ CHỨC CHƯƠNG TRÌNH 9
   2. NGÔN NGỮ CÀI ĐẶT 9
   3. KẾT QUẢ 10
      1. Giao diện chính chương trình 10
      2. Kết quả thực thi chương trình 10
      3. Nhận xét đánh giá 20
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 21
   1. KẾT LUẬN 20
   2. HƯỚNG PHÁT TRIỂN 20

TÀI LIỆU THAM KHẢO 21

**DANH MỤC HÌNH VẼ**

*Hình 2.1: Mô hình mảng một chiều chứa phần tử có kiểu dữ liệu struct* 3

*Hình 2.2: Sơ đồ mô tả quá trình xử lý của một con trỏ* 4

*Hình 3.1: Minh họa thuật toán Bubble Sort* 7

*Hình 4.1: Tổ chức chương trình* 9

*Hình 4.2:* *Giao diện chính của chương trình* 10

*Hình 4.3: Trang lựa chọn ban đầu* 10

*Hình 4.4: Trang đăng nhập* 11

*Hình 4.5: Thông báo đăng nhập thành công* 11

*Hình 4.6: Thông báo sai thông tin tài khoản* 11

*Hình 4.7: Thông báo thoát chương trình do sai tài khoản* 11

*Hình 4.8: Trang đăng ký tài khoản* 11

*Hình 4.9: Thông báo đăng ký tài khoản thành công* 12

*Hình 4.10: Thông báo khi đăng ký không thành công* 12

*Hình 4.11: Trang nhập mã khóa* 12

*Hình 4.12: Trang nhập mã khoa* 13

*Hình 4.13: Trang nhập thông tin sinh viên* 13

*Hình 4.14: Trang nhập thông tin sinh viên (tiếp theo)* 14

*Hình 4.15: File thông tin sinh viên* 14

*Hình 4.16: Thông báo sắp xếp thành công* 14

*Hình 4.17: File thông tin sinh viên sau sắp xếp* 15

*Hình 4.18: Thông báo lớp đã được sắp xếp* 15

*Hình 4.19: Thông báo sắp xếp thất bại* 15

*Hình 4.20: Tìm kiếm sinh viên thành công* 15

*Hình 4.21: Tìm kiếm sinh viên không thành công* 16

*Hình 4.22: Xóa sinh viên thành công* 16

*Hình 4.23: File thông tin sinh viên sau khi xóa* 16

*Hình 4.24: Xóa sinh viên thất bại* 16

*Hình 4.25: Cấp mã số sinh viên thành công* 17

*Hình 4.26: File thông tin sau khi cấp mã số sinh viên* 17

*Hình 4.27: Cấp mã số sinh viên thất bại (do chưa sắp xếp)* 17

*Hình 4.28: Cấp mã số sinh viên thất bại (do lớp không tồn tại)* 17

*Hình 4.29: Cấp địa chỉ email thành công* 18

*Hình 4.30: File thông tin sau khi cấp địa chỉ email* 18

*Hình 4.31: Cấp địa chỉ email thất bại (do chưa cấp mã sinh viên)* 18

*Hình 4.32: Cấp địa chỉ email thất bại (do lớp không tồn tại)* 18

*Hình 4.33: In sinh viên ra file thành công* 19

*Hình 4.34: Thông tin sinh viên trong file report* 19

*Hình 4.35: In sinh viên ra file thất bại* 19

*Hình 4.36: In sinh viên ra màn hình thành công* 19

*Hình 4.37: In sinh viên ra màn hình thất bại* 19

*Hình 4.38: Thoát chương trình* 20

1. **TỔNG QUAN ĐỀ TÀI**

Đề tài: “Xây dựng ứng dụng quản lí danh sách sinh viên” là một đề tài trong lĩnh vực công nghệ thông tin và quản lí phần mềm. Đề tài này tập trung vào xây dựng một ứng dụng quản lí danh sách sinh viên với các chức năng cơ bản: Thêm sinh viên, Sắp xếp danh sách, Tìm kiếm sinh viên, Xóa sinh viên, Cấp mã sinh viên, Cấp email, In danh sách sinh viên.

* 1. **Mục tiêu thực hiện đề tài**
* Mục tiêu chung: Tạo ra một chương trình cung cấp các chức năng quản lí thông tin sinh viên, giúp cho việc quản lí và tra cứu thông tin về sinh viên trở nên dễ dàng hơn, tiết kiệm thời gian và công sức cho người quản lí.
* Mục tiêu cụ thể:

+ Xây dựng hệ thống quản lý thông tin sinh viên với giao diện đơn giản, dễ sử dụng.

+ Cung cấp các chức năng chính trong việc quản lý thông tin sinh viên.

+ Thử nghiệm và đánh giá hiệu quả hệ thống.

* 1. **Phạm vi và đối tượng nghiên cứu:**
* Phạm vi: Tập trung vào việc phát triển một chương trình quản lí thông tin sinh viên cho các trường học, bao gồm các chức năng cơ bản.
* Đối tượng nghiên cứu: Các quá trình quản lí, lưu trữ, thống kê qua đó xây dựng chương trình quản lý sinh viên
  1. **Phương pháp nghiên cứu:**
* Nghiên cứu tài liệu: Tìm kiếm và tham khảo các tài liệu, báo cáo liên quan đến hệ thống quản lý danh sách sinh viên, công nghệ thông tin và lập trình.
* Phân tích hệ thống: Tiến hành phân tích hệ thống quản lý danh sách sinh viên để xác định các hạn chế và nhu cầu cải thiện.
* Thiết kế hệ thống: Phát triển giao diện, mô hình dữ liệu và các chức năng chính của hệ thống.
* Lập trình hệ thống: Xây dựng và triển khai hệ thống quản lý danh sách sinh viên sử dụng ngôn ngữ lập trình C/C++.
* Kiểm thử và hoàn thiện: Thực hiện kiểm thử hệ thống để bảo đảm tính chính xác, hiệu quả, sau đó điều chỉnh và cải tiến dựa trên kết quả kiểm thử.
  1. **Yêu cầu**
* Ngôn ngữ lập trình: Ngôn ngữ lập trình C/C++.
* Cơ sở dữ liệu: Để lưu trữ danh sách sinh viên ta có thể sử dụng các file txt để lưu trữ thông tin danh sách sinh viên và khi cần sử dụng thì ta cần phải đọc file và xử lí nó bằng các thuật toán phù hợp.
* Giao diện người dùng: Ở dự án này ta có thể sử dụng màn hình console để hiển thị giao diện mà ta thao tác sử dụng.
* Thuật toán sắp xếp: Để sắp xếp danh sách sinh viên, ta có thể sử dụng các thuật toán sắp xếp như QuickSort, BubbleSỏt hoặc MergeSort.
* Tạo mã sinh viên và email: Tạo ra một mã sinh viên cho mỗi sinh viên dựa trên danh sách đã được sắp xếp, email sẽ được cấp tự động sau khi cấp mã sinh viên

1. **CƠ SỞ LÝ THUYẾT**
   1. **Ý tưởng**

* Tạo cấu trúc dữ liệu để lưu trư thông tin của sinh viên, bao gồm các trường như: Họ và tên lót, Tên, Ngày sinh, Giới tính, Địa chỉ, Mã sinh viên, Email.
* Tạo một danh sách để lưu trữ danh sách sinh viên. Mỗi phần tử trong danh sách sẽ chứa thông tin của một sinh viên.
* Xây dựng một vòng lặp vô hạn để hiển thị Menu và xử lí lựa chọn của người dùng cho đến khi người dùng chọn thoát (lựa chọn 0).
* Hiển thị Menu cho người dùng và yêu cầu người dùng nhập lựa chọn của mình.
  1. **Cơ sở lý thuyết**

Cơ sở lý thuyết cho đề tài “Xây dựng ứng dụng quản lí danh sách sinh viên” là:

* + 1. **Kiểu dữ liệu struct**
* **Định nghĩa:** Kiểu dữ liệu cấu trúc (struct) là một nhóm các thành phần dữ liệu, được gom lại với nhau và đặt trong một tên. Trong đó:

**+** Mỗi thành phần dữ liệu gọi là **trường dữ liệu** hoặc **thành viên**.

**+** Các **trường dữ liệu** có thể có kiểu dữ liệu khác nhau.

**+** Lập trình viên **tự định nghĩa kiểu dữ liệu struct** gồm những thành phần dữ liệu

* **Cú pháp tổng quát:**

**struct** DataName

{

int data1;

int data2;

…;

};

* + 1. **Mảng một chiều**
* **Định nghĩa:** Mảng (Array) trong C là một tập hợp các phần tử có cùng kiểu dữ liệu và các phần tử được lưu trữ trong một dãy các ô nhớ liên tục trên bộ nhớ. Các phần tử của mảng được truy cập bằng cách sử dụng “chỉ số”. Mảng có kích thước N sẽ có chỉ số từ ) tới N-1.
* **Cú pháp tổng quát:**

type array\_name[array\_size];

Trong đó: - type là kiểu dữ liệu của các phần tử trong mảng.

* array\_name là tên của mảng.
* array\_size là kích thước của mảng vừa được khởi tạo.
  + 1. **Mảng một chiều chứa phần tử có kiểu dữ liệu struct**
* Mảng là tập hợp các phần tử có cùng kiểu dữ liệu với nhau. Các phần tử đó hoàn toàn có thế là các phần tử có kiểu dữu liệu struct.
* Khai báo một mảng cấu trúc cũng giống như khai báo một mảng các kiểu cơ bản. Trong một mảng cấu trúc, mỗi phần tử của mảng đều có kiểu cấu trúc.
* Ví dụ minh họa: Khai báo mảng gồm 10 phần tử có kiểu dữ liệu **struct** Data :

**struct** Data

{

int data1;

int data2;

int data3;

};

typedef Data data;

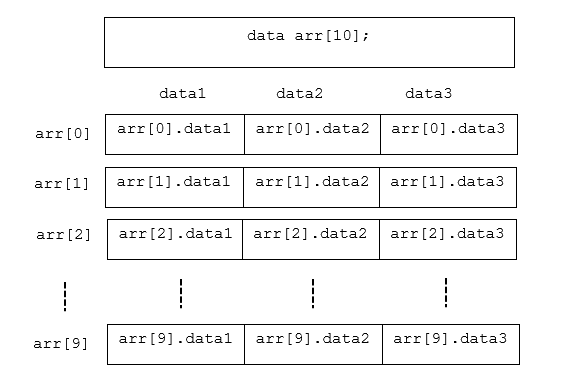
int main()

{

data arr[10];

}

*Hình 2.1: Mô hình mảng một chiều chứa phần tử có kiểu dữ liệu struct*



* + 1. **Con trỏ**
* **Định nghĩa:** Con trỏ là một biến lưu trữ đĩa chỉ của một biến khác. Thay vì lưu trữ một giá trị cụ thể, nó lưu trữ địa chỉ bộ nhớ nơi giá trị đó được lưu trữ. Con trỏ rất hữu ích cho việc quản lý bộ nhớ động, truyền dữ liệu giữa các hàm và truy cập các cấu trúc dữ liệu phức tạp.
* **Cú pháp tổng quát:**

Khai báo con trỏ: **type \*pointer\_name;**

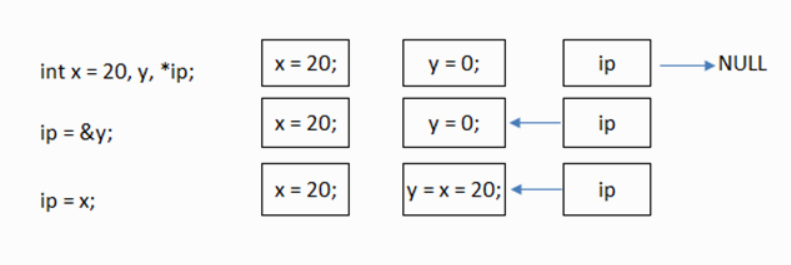
Trong đó: - type là kiểu dữ liệu của biến mà con trỏ sẽ trỏ đến.

* pointer\_name là tên của con trỏ.

Gán địa chỉ cho con trỏ: **pointer\_name = &variable\_name;**

Trong đó: variable\_name là tên của biến mà chúng ta muốn con trỏ trỏ đến.

Truy xuất giá trị thông qua con trỏ: **\*pointer\_name;**



*Hình 2.2: Sơ đồ mô tả quá trình xử lý của một con trỏ*

* + 1. **Chuỗi và các hàm xử lý chuỗi (xâu kí tự)**
* **Định nghĩa:** Chuỗi trong C là một mảng kí tự kết thúc bằng kí tự NULL ‘\0’. Các hàm xử lý chuỗi giúp thực hiện các thao tác như xác định độ dài, sao chép, nối,.. Thư viện chuẩn **<string.h>** cung cấp nhiều hàm hỗ trợ xử lý chuỗi.
* **Cú pháp tổng quát:**

char array\_name[array\_size];

Trong đó: - char là kiểu dữ liệu từng phần tử trong xâu kí tự.

* array\_name là tên của xâu kí tự.
* array\_size là kích thước của xâu kí tự, là số lượng kí tự mà xâu kí tự chứa, bao gồm cả kí tự NULL ‘\0’.

1. **Xác định độ dài chuỗi:** Hàm **strlen** dùng để xác định độ dài chuỗi kí tự (không bao gồm kí tự NULL kết thúc chuỗi).

**Cú pháp:** size\_t strlen(const char \*str);

1. **Sao chép chuỗi:** Hàm **strcpy** dùng để sao chép toàn bộ kí tự của chuỗi này sang chuỗi khác

**Cú pháp:** char \*strcpy(char \*str1, char \*str2);

1. **Nối chuỗi:** Hàm **strcat** dùng để nối chuỗi này vào cuối chuỗi khác, bao gồm cả kí tứ NULL kết thúc chuỗi.

**Cú pháp:** char \*strcat(char \*dest, const char \*src);

1. **So sánh hai chuỗi:** Hàm **strcmp** dùng để so sánh hai chuỗi với nhau. Hàm sẽ so sánh hai chuỗi và trả về các giá trị tương ứng.

**Cú pháp:** int strcmp(const char \*str1, const char \*str2);

* + 1. **File và các thuật toán thao tác với file**
* **Định nghĩa:** Tệp (File) là một vùng chứa trong các thiết bị lưu trữ của máy tính được sử dụng để lưu trữ dữ liệu. Một file biểu diễn chuỗi các bytes, không kể đó là file văn bản hay file nhị phân. Ngôn ngữ lập trình C cung cấp các hàm truy cập mức độ cao cũng như thấp để thao tác với file trên thiết bị lưu trữ.

1. **Mở file:** Hàm **fopen** dùng để mở file đã tồn tại hoặc cũng có thể dùng để tạo file mới.

**Cú pháp:**

FILE \*fp = fopen(const char \*filename, const char mode);

1. **Đóng file:** Hàm **fclose** dùng để đóng lại file đã được mở bằng hàm **fopen**

**Cú pháp:** int fclose(FILE \*fp);

1. **Ghi vào file:** Hàm **fprint** dùng để ghi dữ liệu vào file.

**Cú pháp:** int fprintf(FILE \*fp, const char \*format, ...);

1. **Đọc dữ liệu từ file:** Hàm **fscanf** dùng để đọc dữ liệu từ một file đã mở và lưu dữ liệu đó vào các biến.

**Cú pháp:** int fscanf(FILE \*fp, constchar \*format ...);

1. **Đọc một dòng từ file:** Hàm **fgets** dùng để đọc một dòng từ file đã cho và lưu trữ nó vào trong chuỗi.

**Cú pháp:** char\* fgets(char\* str, int n, FILE \*fp);

1. **TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN**
   1. **Phát biểu bài toán**

Đầu vào (Input) của chương trình: Đầu vào nhập từ bàn phím

* Thông tin về sinh viên mới bao gồm Họ và tên lót, Tên, Ngày sinh, Giới tính và Địa chỉ.
* Yêu cầu sắp xếp danh sách.
* Yêu cầu Xóa sinh viên dựa trên Mã sinh viên.
* Yêu cầu tìm kiếm và in ra thông tin sinh viên dựa trên Mã sinh viên.
* Yêu cầu cấp Mã sinh viên.
* Yêu cầu cấp Email.

Đầu ra (Output) của chương trình bao gồm:

* File bao gồm danh sách sinh viên với thông tin về Mã sinh viên, Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Email đã được chỉnh sửa.
* File chứa dữ liệu thông tin sinh viên của các lớp.
  1. **Cấu trúc dữ liệu**

Kiểu dữ liệu struct Student là một cấu trúc dữ liệu dùng để định nghĩa thông tin của các sinh viên, nhóm các biến thông tin này dưới một tên Student duy nhất. Điều này giúp cho việc tham gia quản lý thông tin của các sinh viên một cách hiệu quả và linh hoạt hơn.

struct Student

{

char LastName[Max\_LastName];

char FirstName[Max\_FirstName];

char FullName[Max\_FullName];

char Birthday[Max\_Date];

char Gender[Max\_Gender];

char Address[Max\_Address];

char ID[Max\_ID];

char email[Max\_email];

};

Các thuộc tính bao gồm:

* LastName: Họ và tên lót của sinh viên.
* FirstName: Tên của sinh viên.
* FullName: Họ và tên của sinh viên.
* Birthday: Ngày tháng năm sinh của sinh viên.
* Gender: Giới tính của sinh viên.
* Address: Địa chỉ của sinh viên.
* ID: Mã số sinh viên của sinh viên.
* email: Địa chỉ email của sinh viên.
  1. **Thuật toán**

Thuật toán của bài quản lí danh sách sinh viên được thể hiện qua:

* + 1. **Thuật toán SortStudent (Sắp xếp sinh viên)**

Sử dụng phương pháp sắp xếp nổi bọt (bubble sort):

* Ý tưởng: Thuật toán sắp xếp nổi bọt thực hiện sắp xếp các phần tử bằng cách lặp lại qua mảng, so sánh từng cặp phần tử liền kề và hoán đổi chúng nếu chúng ở sai thứ tự. Quá trình này được lặp lại cho đến khi mảng được sắp xếp
* Cách thực hiện thuật toán Bubble Sort:

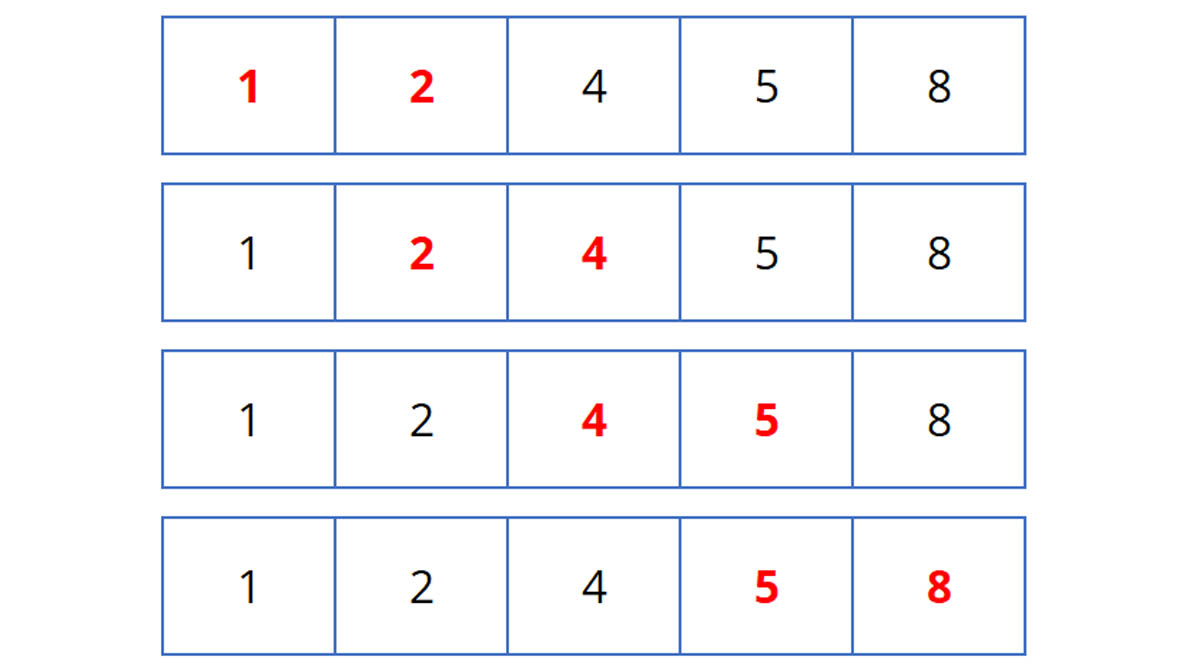
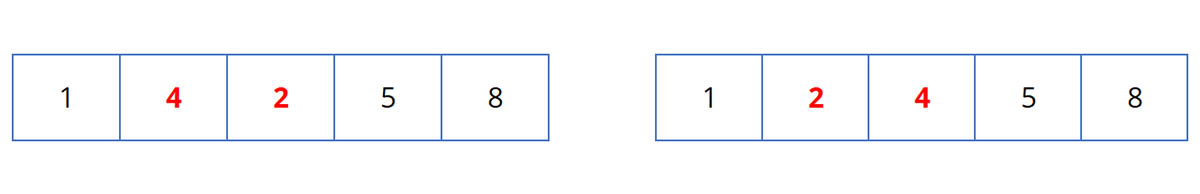
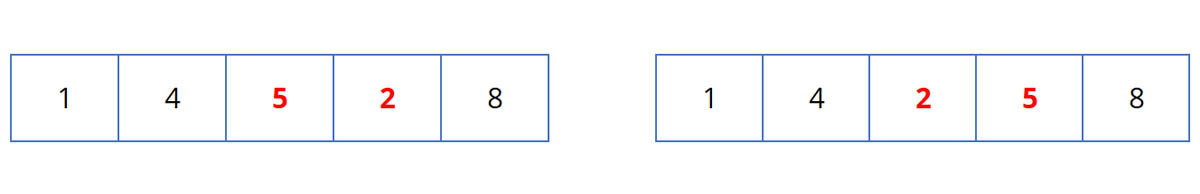
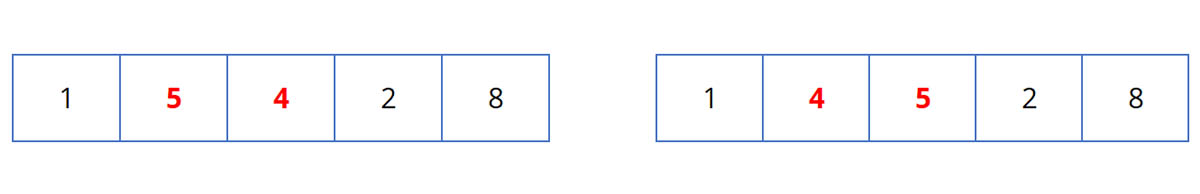
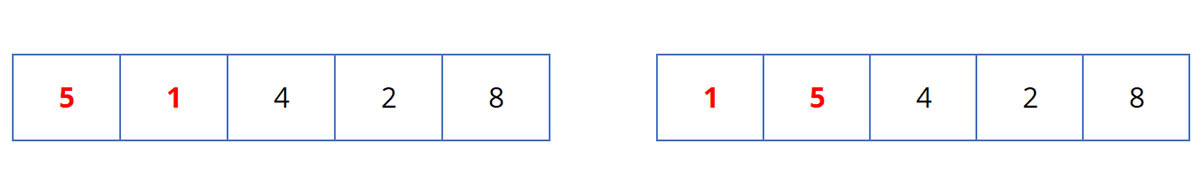
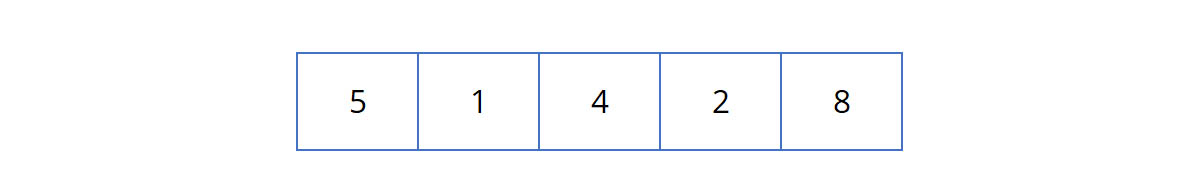
Bước 1: So sánh hai phần tử liền kề đầu tiên của mảng.

Bước 2: Nếu phần tử bên trái lớn hơn phần tử bên phải, hoán đổi chúng.

Bước 3: Di chuyển sang cặp phần tử tiếp theo và lặp lại bước 2 cho đến khi kết thúc mảng. Sau mỗi lần lặp, phần tử lớn nhất sẽ được “bong bóng” đến vị trí cuối cùng của phần chưa được sắp xếp trong mảng.

Bước 4: Giảm kích thước của mảng chưa được sắp xếp đi một và lặp lại quá trình cho đến khi mảng được sắp xếp.

* Ví dụ minh họa:



*Hình 3.1: Minh họa thuật toán BubbleSort*

* + 1. **Thuật toán AddStudent (Thêm sinh viên)**

Bước 1: Mở file lớp cần thêm sinh viên. Nếu lớp chưa tồn tại thì tạo lớp và thêm tên lớp vào file ClassList theo mã khóa và mã khoa, sau đó đóng file.

Bước 2: Nhập thông tin từng sinh viên trong vòng lặp do-while. Sau mỗi lần nhập, hỏi người dùng có muốn tiếp tục nhập thông tin không.

Bước 3: Nếu lớp mới, ghi thông tin sinh viên vào file với dòng đầu là số lượng sinh viên, dòng hai là kiểm tra sắp xếp (‘0’), và các dòng tiếp theo là thông tin sinh viên. Nếu lớp đã sắp xếp thì tiến hành kiểm tra kí tự checksort.

Bước 4: Nếu checksort = 0 thì sẽ thêm sinh viên vào cuối file rồi cập nhật lại số lượng sinh viên trong danh sách. Còn nếu checksort = 1 thì chèn sinh viên được thêm vào vị trí hợp lý và cập nhật lại số lượng sinh viên trong danh sách.

* + 1. **Thuật toán CreateStudentID (Cấp mã sinh viên)**

Bước 1: Đọc dòng thứ nhất của file lưu vào StdCount, đọc dòng thứ hai lưu vào checksort.

Bước 2: Nếu checksort = 0 thì sẽ báo là “Vui lòng sắp xếp danh sách sinh viên trong lớp trước khi cấp mã sinh viên”, và thoát khỏi hàm, trả về Menu, ngược lại nếu checksort = 1 thì tiến hành vòng lặp đến khi kết thúc file lưu vào mảng Student.

Bước 3: Lặp 1 đến StdCount (duyệt hết các sinh viên trong danh sách), nếu như sinh viên đã được cấp mã thì sẽ tiến hành bỏ qua, còn nếu như sinh viên chưa được cấp mã thì sẽ tiến hành cấp mã bằng cách dùng hàm sprintf lưu trữ vào Student[i].ID lần lượt các chuỗi FaculityCode, SchoolYear và 4 chữ số được đọc từ FileID của khoa và khóa được chọn.

* + 1. **Thuật toán CreateEmail (Cấp email)**

Bước 1: Đọc dòng thứ nhất của file lưu vào StdCount, đọc dòng thứ hai lưu vào checksort.

Bước 2: Lặp từ 1 đến StdCount. Nếu có sinh viên chưa cấp mã thì sẽ báo “Vui lòng cấp mã sinh viên trước khi cấp email và thoát khỏi hàm.

Bước 3: Lặp từ 1 đến StdCount. Nếu như sinh viên đã có email thì sẽ tiến hành bỏ qua, còn nếu chưa có email thì tiến hành sao chép StudentList[i].ID vào StudentList[i[.email bằng hàm strcpy, sau đó nối chuỗi “@sv.dut.udn.vn” vào StudentList[i].email bằng hàm strcat.

* + 1. **Thuật toán FindStudent (Tìm kiếm sinh viên)**

Bước 1: Lặp cho đến khi kết thúc file các lớp theo mã khoa và mã khóa, mỗi lần đọc lưu vào ClassFindName và tiến hành các bước tiếp theo.

Bước 2: Đọc dòng thứ nhất của file ClassindName lưu vào StdCount, đọc dòng thứ hai lưu vào checksort. Lặp đến khi kết thúc file và lưu từng dòng vào StudentList.

Bước 3: Khai báo found = false. Lặp từ 1 đến StdCount , nếu như có sinh viên có mã sinh viên giống với mã sinh viên cần tìm thì gán found = true, in các thông tin Họ và tên, Ngày sinh, Giới tính, Địa chỉ ra màn hình và dừng vòng lặp tìm kiếm, ngược lại thì tiến hành duyệt sang lớp tiếp theo.

Bước 4: Nếu found = true sẽ dừng vòng lặp đọc tên lớp và thoát khỏi hàm.

Bước 5: Nếu đọc tất cả lớp mà không có sẽ thông báo không có sinh viên có mã sinh viên cần tìm.

* + 1. **Thuật toán RemoveStudent (Xóa sinh viên)**

Bước 1: Lặp cho đến khi kết thúc file các lớp theo mã khoa và mã khóa, lưu một dòng vào ClassFindName và tiến hành các bước tiếp theo.

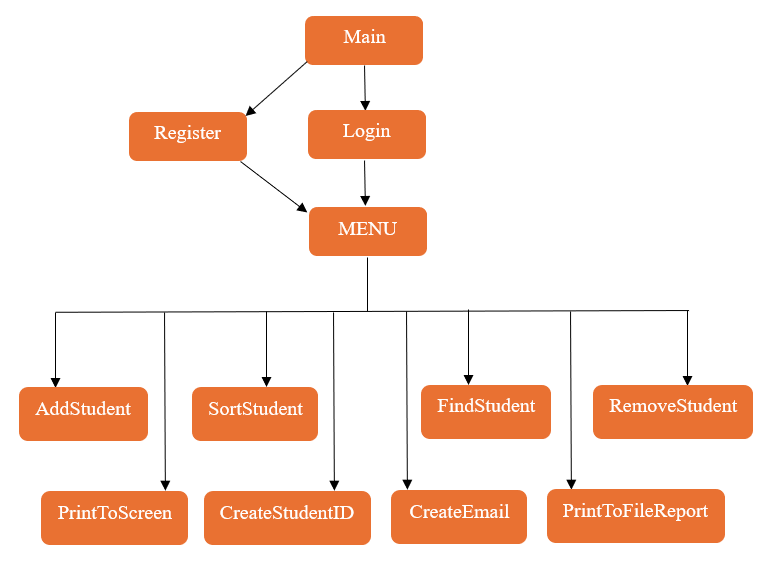
Bước 2: Đọc dòng thứ nhất của file ClassindName lưu vào StdCount, đọc dòng thứ hai lưu vào checksort. Lặp đến khi kết thúc file và lưu từng dòng vào StudentList.

Bước 3: Khai báo found = false. Lặp từ 1 đến StdCount, nếu như có sinh viên có mã sinh viên tại vị trí I giống với mã sinh viên cần xóa thì lặp từ i đến StdCount, dịch chuyển các sinh viên ở sau lên một vị trí, giảm StdCount đi 1, gán found = true và dừng vòng lặp.

Bước 4: Nếu giá trị found là true thì sẽ tiến hành cập nhật lại danh sách lớp có sinh viên vừa được xóa và thoát khỏi hàm

Bước 5: Nếu đọc tất cả lớp mà không có sẽ thông báo không có sinh viên có mã sinh viên cần xóa và thoát khỏi hàm.

1. **CHƯƠNG TRÌNH VÀ KẾT QUẢ**
   1. **Tổ chức chương trình**



*Hình 4.1: Tổ chức chương trình*

* 1. **Xây dựng chương trình**

Bài toán quản lí danh sách sinh viên là một trong những bài toán phổ biến trong lập trình, được thực hiện bằng nhiều ngôn ngữ khác nhau. Trong trường hợp này, chúng ta sử dụng ngôn ngữ C để triển khai chương trình.

#include <stdio.h>

#include <string.h>

#include <windows.h>

#include <stdbool.h>

#include <conio.h>

#include <ctype.h>

#define MAX\_SIZE 100

#define Max\_Class 30

#define Max\_LastName 30

#define Max\_FirstName 10

#define Max\_FullName 30

#define Max\_Date 15

#define Max\_Gender 10

#define Max\_Address 30

#define Max\_FaculityCode 5

#define Max\_SchoolYear 5

#define Max\_ID 20

#define Max\_email 30

#define Max\_Username 30

#define Max\_Password 30

typedef struct

{

char Username[Max\_Username];

char Password[Max\_Password];

} USER;

struct Student

{

char LastName[Max\_LastName];

char FirstName[Max\_FirstName];

char FullName[Max\_FullName];

char Birthday[Max\_Date];

char Gender[Max\_Gender];

char Address[Max\_Address];

char ID[Max\_ID];

char email[Max\_email];

};

typedef struct Student STUDENT;

STUDENT Student[MAX\_SIZE];

char FaculityCode[Max\_FaculityCode];

char SchoolYear[Max\_SchoolYear];

char filename[30];

char ClassName[15];

char ClassFindName[MAX\_SIZE];

int StdCount = 0;

char checksort;

void RemoveEnter(char x[]);

void MakeNULL(char x[]);

char \*rTrim(char x[]);

char \*lTrim(char x[]);

char \*trim(char x[]);

char \*NameStr(char x[]);

void CreateClass(char filename[]);

void SwapStudent(STUDENT \*student1, STUDENT \*student2);

void BubbleSort(STUDENT Student[], int StdCount);

void ReadStudentFile(FILE \*file, int \*StdCount);

void PrintToFile(FILE \*file, STUDENT Student[], int StdCount);

void Input\_Student(STUDENT \*student);

void AddStudent();

void SortStudent();

void CreateStudentID();

void CreateEmail();

void FindStudent();

void RemoveStudent();

void PrintToFileReport();

void PrintToScreen();

void Start();

bool Login();

bool CheckRegister(char username[]);

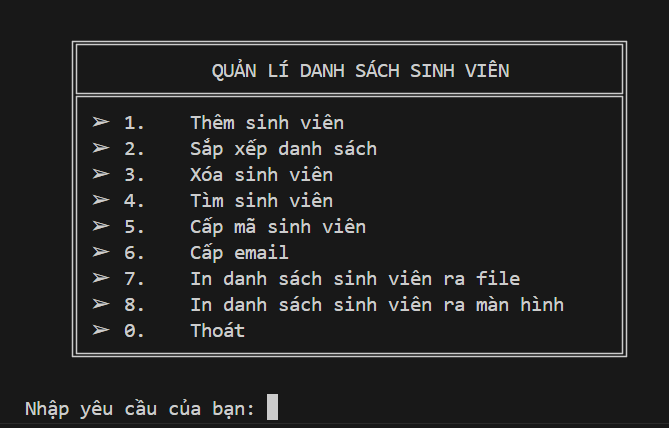
void Register();

void MENU();

* 1. **Kết quả** 
     1. **Giao diện chính của chương trình**

Giao diện chính của chương trình cung cấp các lựa chọn cụ thể giúp người dùng dễ dàng chọn yêu cầu mong muốn.

*Hình 4.2: Giao diện chính của chương trình*



* + 1. **Kết quả thực thi chương trình**

1. **Chương trình đăng nhập / đăng kí tài khoản**

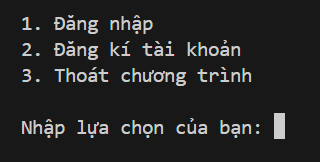
Chương trình sẽ hiển thị ra ba lựa chọn sau:

+ Một là đăng nhập vào để sử dụng chương trình quản lí sinh viên.

+ Hai là đăng kí tài khoản mới để sử dụng chương trình.

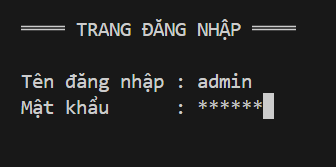
+ Ba là thoát khỏi chương trình nếu chưa muốn vào chương trình.

*Hình 4.3: Trang lựa chọn ban đầu*



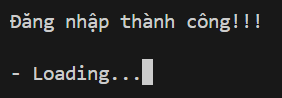
Với lựa chọn 1 sẽ đưa đến trang đăng nhập.

*Hình 4.4: Trang đăng nhập*



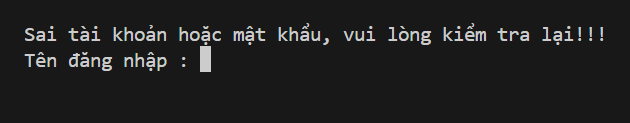
Nếu như nhập đúng tài khoản thì sẽ đước báo là đăng nhập thành công và tiến hành vào sử dụng chương trình.

*Hình 4.5: Thông báo đăng nhập thành công*



Nếu như sai tài khoản hoặc mật khẩu thì sẽ thông báo sai và tiến hành nhập lại.

*Hình 4.6: Thông báo sai thông tin tài khoản*



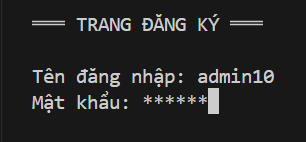
Khi nhập sai tài khoản hoặc mật khẩu quá 3 lần thì sẽ tiến hành thoát chương trình.

*Hình 4.7: Thông báo thoát chương trình do sai tài khoản*



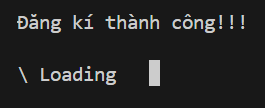
Với lựa chọn 2 sẽ đưa đến trang đăng kí tài khoản.

*Hình 4.8: Trang đăng ký tài khoản*



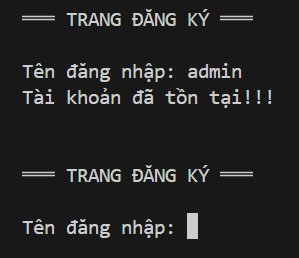
Sau khi nhập vào tài khoản và mật khẩu, nếu thõa mản điều kiện thì sẽ báo là đăng kí tài khoản thành công và chuyển đến sử dụng chương trình.

*Hình 4.9: Thông báo đăng kí tài khoản thành công*



Nếu như nhập vào tên đăng nhập mà đã có tài khoản có tên đăng nhập này thì sẽ thông báo là tài khoản đã tồn tại và đưa về lại trang đăng kí.

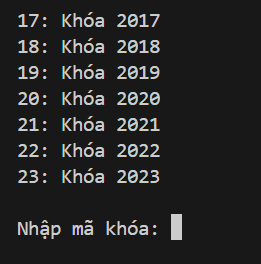
*Hình 4.10: Thông báo khi đăng ký không thành công*



1. **Chức năng nhập mã khoa, nhập mã khóa**

Sau khi đăng nhập (đăng kí) thành công thì sẽ dẫn đến sử dụng chương trình. Trước tiên sẽ nhập mã khóa với các gợi ý như sau:

*Hình 4.11: Trang nhập mã khóa*



Sau khi nhập xong mã khóa thì sẽ đến mã khoa, lúc này cũng sẽ có các gợi ý như sau và sẽ tiến hành nhập mã:

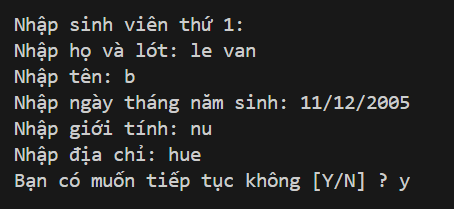
*Hình 4.12: Trang nhập mã khoa*



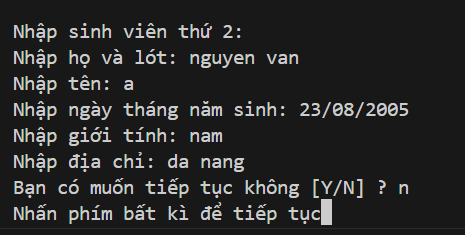
1. **Chức năng thêm sinh viên**

Với chức năng thêm sinh viên, ban đầu sẽ tiến hành nhập tên lớp cần thêm sinh viên, sau đó sẽ tiến hành điền các thông tin Họ và tên lót, Tên, Ngày tháng năm sinh, Giới tính, Địa chỉ.

*Hình 4.13: Trang nhập thông tin sinh viên*



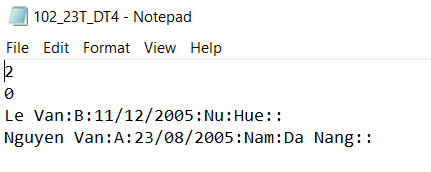
Sau khi nhập xong thông tin của sinh viên đầu tiên sẽ được hỏi rằng “Bạn có muốn tiếp tục không [Y/N]?” với 2 đáp án trả lời là Y và N. Nếu như nhập Y sẽ tiến hành sang nhập thông tin sinh viên thứ 2.



*Hình 4.14: Trang nhập thông tin sinh viên (tiếp theo)*

Ngược lại, nếu như trả lời là N thì sẽ dừng việc thêm sinh viên và trả về Menu để nhập lựa chọn tiếp theo.

Sau khi thêm sinh viên thì sẽ thêm vào danh sách lớp như trên. Dòng đầu riên sẽ chứa số lượng sinh viên, dòng thứ 2 sẽ chứa kí tự kiểm tra sắp xếp sau khi thêm sinh viên sẽ được khởi tạo là 0 (chưa sắp xếp). Các thông tin được cách nhau bởi dấu : mỗi lần đọc sẽ truy cập lần lượt Họ và tên lót, Tên, Ngày tháng năm sinh, Giới tính, Địa chỉ, Mã sinh viên và Email (nếu chưa có sẽ để trống).

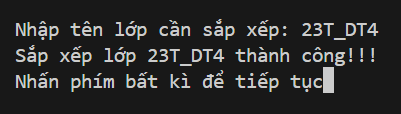


*Hình 4.15: File thông tin sinh viên*

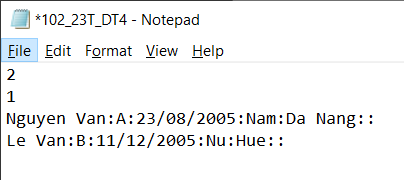
1. **Chức năng sắp xếp danh sách**

Với chức năng sắp xếp danh sách, ban đầu sẽ tiến hành nhập vào tên lớp cần sắp xếp danh sách sinh viên. Lúc này sẽ tiến hành mở file lớp đọc thông tin. Nếu như lớp chưa được sắp xếp sẽ tiến hành sắp xếp.

*Hình 4.16: Thông báo sắp xếp thành công*

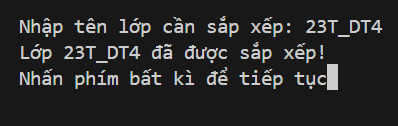


Sau khi sắp xếp thì sẽ tiến hành khởi tạo kí tự kiểm tra sắp xếp thành 1.



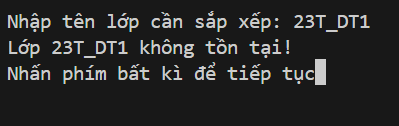
*Hình 4.17: File thông tin sinh viên sau sắp xếp*

Nếu như lớp đã được sắp xếp rồi thì sẽ báo là đã được sắp xếp.



*Hình 4.18: Thông báo lớp đã được sắp xếp*

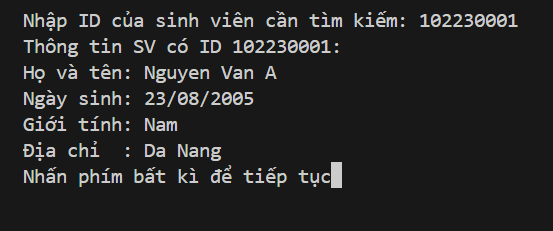
Nếu như lớp không có sinh viên nào sẽ báo là lớp này chưa tồn tại.



*Hình 4.19: Thông báo sắp xếp thất bại*

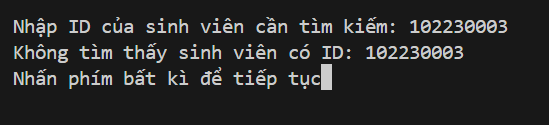
1. **Chức năng tìm kiếm sinh viên**

Với chức năng này ban đầu sẽ yêu cầu nhập mã sinh viên của sinh viên cần tìm kiếm.



*Hình 4.20: Tìm kiếm sinh viên thành công*

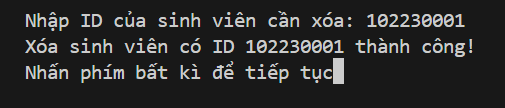
Nếu như không có mã sinh viên thỏa mãn thì sẽ thông báo như sau và trả về Menu.



*Hình 4.21: Tìm kiếm sinh viên không thành công*

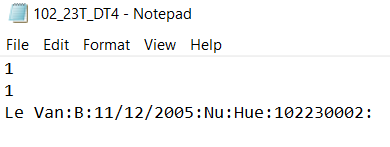
1. **Chức năng xóa sinh viên**

Với chức năng này ban đầu sẽ yêu cầu nhập mã sinh viên của sinh viên cần xóa. Nếu như tìm thấy thì sẽ thông báo xóa thành công.



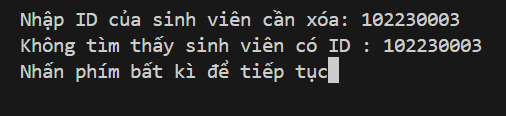
*Hình 4.22: Xóa sinh viên thành công*

Lúc này danh sách trong lớp sẽ được cập nhật lại.



*Hình 4.23: File thông tinh sinh viên sau khi xóa*

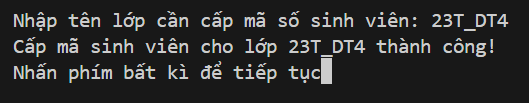
Nếu như không tìm thấy sinh viên cần xóa thì sẽ thông báo không tìm thấy và trả về Menu.



*Hình 4.24: Xóa sinh viên thất bại*

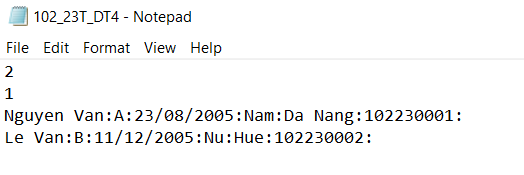
1. **Chức năng cấp mã sinh viên**

Với chức năng này, ban đầu ta sẽ yêu cầu nhập tên lớp cần cấp mã số sinh viên. Nếu như sinh viên đã được cấp thì tiến hành bỏ qua không cần cấp lại. Lúc này sẽ thông báo trên màn hình là cấp mã số sinh viên thành công.



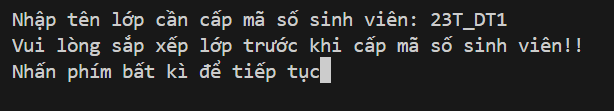
*Hình 4.25: Cấp mã số sinh viên thành công*

Sau khi thực hiện yêu cầu cấp mã số sinh viên thì thông tin trong file sẽ được cập nhật lại với việc bổ sung thêm mã số sinh viên.



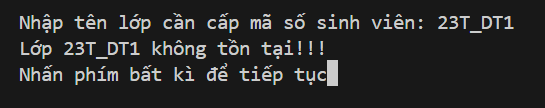
*Hình 4.26: File sau khi cấp mã sinh viên*

Nếu như lớp chưa được sắp xếp thì khi tiến hành cấp mã sẽ được báo là phải sắp xếp lớp trước khi cấp mã. Cần phải tiến hành sắp xếp lớp mới có thể cấp mã số sinh viên.



*Hình 4.27: Cấp mã số sinh viên thất bại (do chưa sắp xếp)*

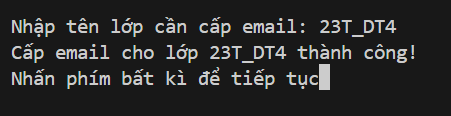
Nếu như lớp chưa được khởi tạo thì sẽ thông báo là lớp không tồn tại, không thể tiến hành cấp mã số sinh viên.



*Hình 4.28: Cấp mã số sinh viên thất bại (do lớp không tồn tại)*

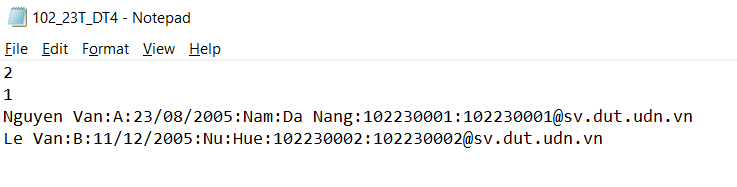
1. **Chức năng cấp email**

Với chức năng này, ban đầu ta sẽ yêu cầu nhập tên lớp cần cấp mã địa chỉ email. Nếu như sinh viên đã được cấp thì tiến hành bỏ qua không cần cấp lại. Lúc này sẽ thông báo trên màn hình là cấp email thành công.



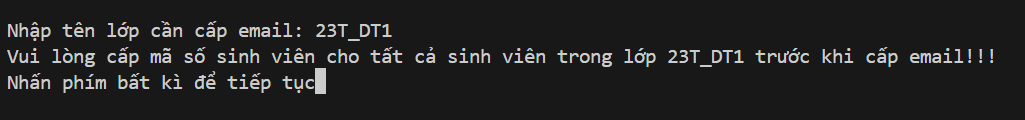
*Hình 4.29: Cấp email sinh viên thành công*

Sau khi thực hiện yêu cầu cấp mã số sinh viên thì thông tin trong file sẽ được cập nhật lại với việc bổ sung thêm địa chỉ email.

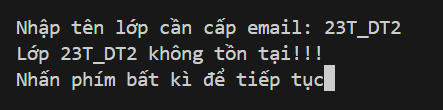


*Hình 4.30: File sau khi cấp email*

Nếu như có một sinh viên trong lớp chưa được cấp mã số sinh viên thì việc cấp mã số sinh viên sẽ không thể diễn ra. Cần phải cấp mã sinh viên trước khi cấp địa chỉ email.



*Hình 4.31: Cấp email sinh viên thất bại (do chưa cấp mã sinh viên)*

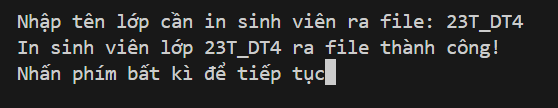


*Hình 4.32: Cấp email sinh viên thất bại (do lớp không tồn tại)*

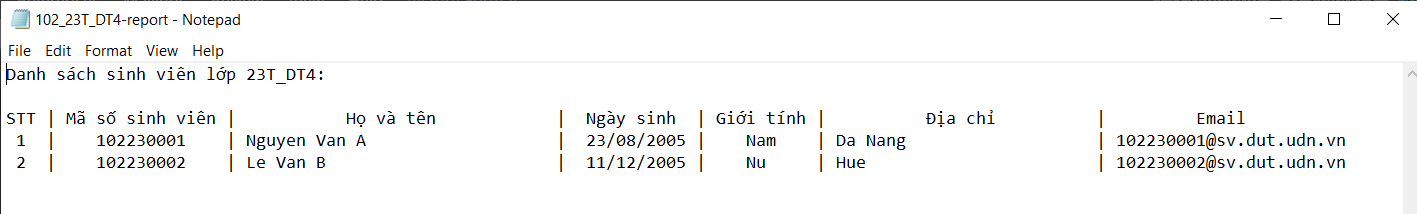
1. **Chức năng in danh sách sinh viên ra file**

Với chức năng này, ta sẽ được yêu cầu nhập tên lớp cần in thông tin sinh viên ra file. File sẽ được lưu dưới tên “mã khoa\_tên lớp\_report”. Lúc này màn hình sẽ thông báo in thành công.

*Hình 4.33: In sinh viên ra file thành công*



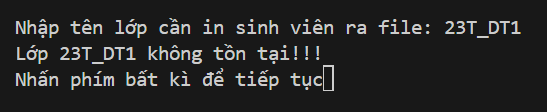
Khi in thành công thì file report của lớp sẽ được khởi tạo gồm thông tin của các sinh viên trong lớp được biểu diễn đẹp mắt.



*Hình 4.34: Thông tin sinh viên trong file report*

Nếu như lớp chưa được khỏi tạo thì sẽ thông báo in sinh viên ra file thất bại.

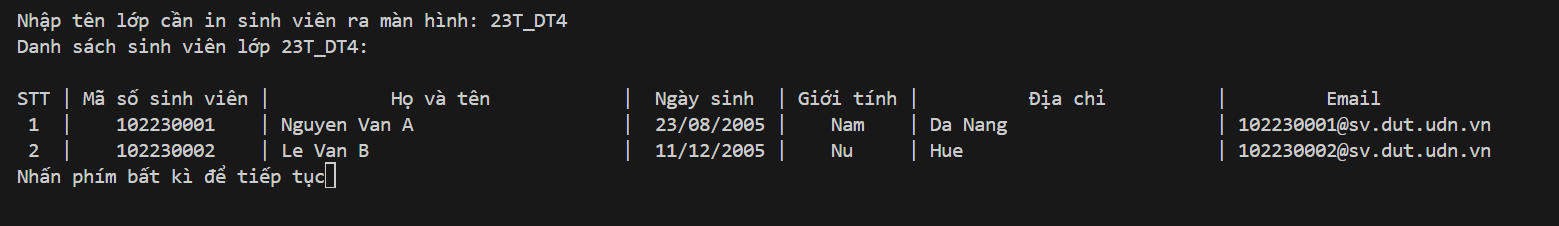
*Hình 4.35: In sinh viên ra file thất bại*



1. **Chức năng in danh sách sinh viên ra màn hình**

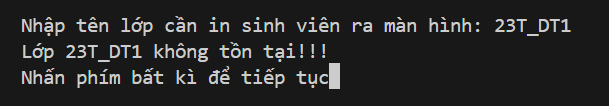
Với chức năng này, ta sẽ được yêu cầu nhập tên lớp cần in thông tin sinh viên ra màn hình. Lúc này màn hình sẽ hiện ra thông tin sinh viên của lớp được yêu cầu.

*Hình 4.36: Thông tin sinh viên trên màn hình*



Nếu như lớp không tồn tại thì sẽ thông báo in sinh viên ra màn hình thất bại.

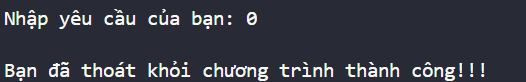
*Hình 4.37: In thông tin sinh viên thất bại*



1. **Chức năng thoát**

Dùng để thoát chương trình.

*Hình 4.38: Thông báo thoát chương trình*



* + 1. **Nhận xét đánh giá**
* **Tích cực:**

+ Chương trình đã đáp ứng đầy đủ các yêu cầu để thực hiện quản lí danh sách sinh viên của các lớp, đã xây dựng được một chương trình quản lý sinh viên mang tính chất khách quan.

+ Việc cài đặt dễ dàng khi nắm được kiến thức cơ bản về cấu trúc dữ liệu struct, mảng, con trỏ…

+ Chương trình có hiệu suất tốt, thực hiện thao tác nhanh chóng và mượt mà, có giao diện người dùng dễ sử dụng, cung cấp các tùy chọn rõ ràng để thao tác.

* **Hạn chế:**

+ Giao diện chưa có tính thẩm mĩ cao, màu sắc, kích thước của các dòng làm người dùng có thể khó quan sát.

+ Chương trình chưa tối ưu và chưa có nhiều tiện ích.

+ Tốc độ xử lí chưa được nhanh chóng.

1. **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**
   1. **Kết luận**

Tổng quan lại, bài toán quản lí danh sách sinh viên là bài toán mặc dù đơn giản nhưng lại vô cùng hữu ích, tiện lợi trong rất nhiều trường hợp và đặc biệt quan trọng trong việc quản lí danh sách sinh viên. Với chương trình này, người dùng có thể dễ dàng sử dụng và tiến hành các thao tác mong muốn đối với danh sách sinh viên, đảm bảo các thao tác sẽ được như ý muốn, đầy đủ tính chính xác và tính hiệu quả trong việc quản lí. Chương trình này có thể được sử dụng rộng rãi cho việc quản lí học sinh, sinh viên ở các trường đại học trong cả nước.

* 1. **Hướng phát triển**

Mặc dù chương trình đã đáp ứng đủ yêu cầu cơ bản nhưng vẫn cần bổ sung thêm một số tính năng khác để nó có thể trở nên phù hợp với mọi người quản lí.

* Có thể thêm các chức năng khác như: sao lưu, phục hồi dữ liệu… cho chương trình
* Từ đây xây dựng được hệ thống quản lí thư viện giúp quản lí việc mượn sách của sinh viên dễ dàng hơn.
* Thêm chức năng thông báo của nhà trường từ việc kết hợp việc quản lí điểm, tiến độ học tập của sinh viên và gửi thông báo về sinh viên.

**TÀI LIỆU THAM KHẢO**

*(1): Tài liệu về strings và file của Ths. Trần Hồ Thủy Tiên (tài liệu lưu hành nội bộ).*

*(2): Mảng một chiều trong C* [Mảng](https://blog.28tech.com.vn/c-mang-1-chieu)

*(3): Con trỏ trong C* [Con trỏ](https://teky.edu.vn/blog/con-tro-trong-c/)

*(4): Xâu kí tự trong C* [Chuỗi](https://viblo.asia/p/bai-10-xau-ki-tu-gDVK2rX0KLj)