**CHƯƠNG 1: GIỚI THIỆU**

**1. Tổng quan**

Hiện nay, sự phát triển của Công nghệ thông tin ở nước ta đang bước vào một kỷ nguyên mới với việc triển khai rộng rãi các ứng dụng tin học cho các tổ chức và xã hội. Không còn nghi ngờ gì nữa về vai trò của Công nghệ thông tin đối với đời sống, đối với khoa học kỹ thuật, trong kinh doanh, cũng như mọi mặt của xã hội, ngay cả đối với một cá nhân.

Tuy nhiên, ở thời điểm hiện tại, Công nghệ thông tin mới chỉ bước đầu được ứng dụng trong cuộc sống nói chung và giáo dục nói riêng. Đối với các trường đại học lớn, việc quản lý sinh viên với hàng nghìn hồ sơ và quản lý việc nộp, rút hồ sơ của hàng nghìn sinh viên là vô cùng phức tạp. Vì vậy, em đã nhận đề tài nghiên cứu: “Xây dựng chương trình quản lý hồ sơ sinh viên”

**2. Tầm quan trọng của đề tài**

Để giải quyết phần nào những hạn chế của quản lý hồ sơ sinh viên thủ công, chúng ta phải công nghệ hóa các công việc quản lý hồ sơ sinh viên. Nhằm phục vụ tốt hơn cho lượng hồ sơ sinh viên ngày càng tăng. Vì vậy, một hệ thống quản lý hồ sơ sinh viên phải thực hiện các chức năng chính như: tra cứu, nhập dữ liệu sinh viên, cập nhật thông tin, ...

**3. Mục tiêu của đề tài**

Chương trình này sẽ được sử dụng trong các trường đại học. Nhân viên của trường sẽ sử dụng hệ thống để thêm, sửa, tìm kiếm, sắp xếp hồ sơ của sinh viên và đọc từ file cũng như lưu thông tin của sinh viên vào file.

**CHƯƠNG 2: CÀI ĐẶT VÀ KIỂM THỬ**

**1. Cấu trúc của chương trình**

Chúng ta sử dụng struct để định nghĩa một cấu trúc Student, một mảng struct để lưu trữ listStudents, và sau đây là các bước cụ thể để tạo một chương trình Student management trong C bằng struct và File thỏa mãn bốn chức năng: add, sort, update, search.

**Tạo cấu trúc Student:**

Chúng ta sẽ tạo cấu trúc sinh viên như sau: name, birthday, course, class được khai báo với kiểu dữ liệu char còn studentID thì được khai báo với kiểu dữ liệu int . Ngoài ra, chúng ta còn khai báo biến toàn cục listStudents và biến numStudents được gán với giá trị bằng 0.

**Hàm readFile():**

Trước tiên, ta cần khai báo con trỏ kiểu FILE để nó trỏ đến file “ Student.txt” cần đọc trên máy tính và dùng hàm fopen() để mở file và mode “ r” là mở để đọc .Sau đó , sử dụng câu lệnh if và khai báo các biến ra bên ngoài bằng fscanf .

Sau khi mở file và thực hiện xong thì ta đóng file lại bằng cách sử dụng hàm fclose (fOut) , trong đó fOut là con trỏ của file đang xử lý .

**Hàm writeFile() :**

Chúng ta tiếp tục khai báo con trỏ kiểu File nhưng mode “ w+” là mở để đọc và ghi . Khai báo biến i và sử dụng vòng lặp for để ghi tất cả dữ liệu trong file . Và không quên đóng file sau khi thực hiện xong các câu lệnh .

**Hàm add Student ():**

Khai báo biến student thuộc kiểu dữ cấu trúc Student và sử dụng printf , gets để nhập và xuất tất cả dữ liệu được đưa vào từ bàn phím .

**Hàm updateStudentByID ():**

Trước tiên , ta cần khai báo các mảng 1 chiều mới : tempName , tempBirthday , tempCourse, tempclass để phục vụ cho thuật toán chỉnh sửa . Tiếp tục sử dụng vòng lặp for và câu lệnh điều kiện if để xác định student cần chỉnh sửa và thực hiện thuật toán chỉnh sửa .

**Hàm searchStudent():**

Khai báo các mảng 1 chiều : tempName , tempClass với kiểu dữ liệu char và id với kiểu dữ liệu int để phục vụ cho thuật toán tìm kiếm . Sử dụng hàm “ switch – case “ với 3 case là thực hiện tìm kiếm bằng mã sinh viên , họ tên và lớp .

Trong case 1, sử dụng kết hợp vòng lặp for và câu lệnh điều kiện if để kiểm tra id nhập vào có khớp với các id có trong danh sách , nếu trùng thì xuất ra màn hình các thông tin của sinh viên đó bằng hàm printf , ngược lại thì xuất “ Can found student with id “.

Đối với case 2 và case 3, ta thực hiện tương tự như case 1, đều sử dụng vòng lặp for và câu lệnh điều kiện if để thực hiện thuật toán tìm kiếm .

Trong trường hợp nhập sai case thì in ra màn hình “ Wrong function , please choose again”.

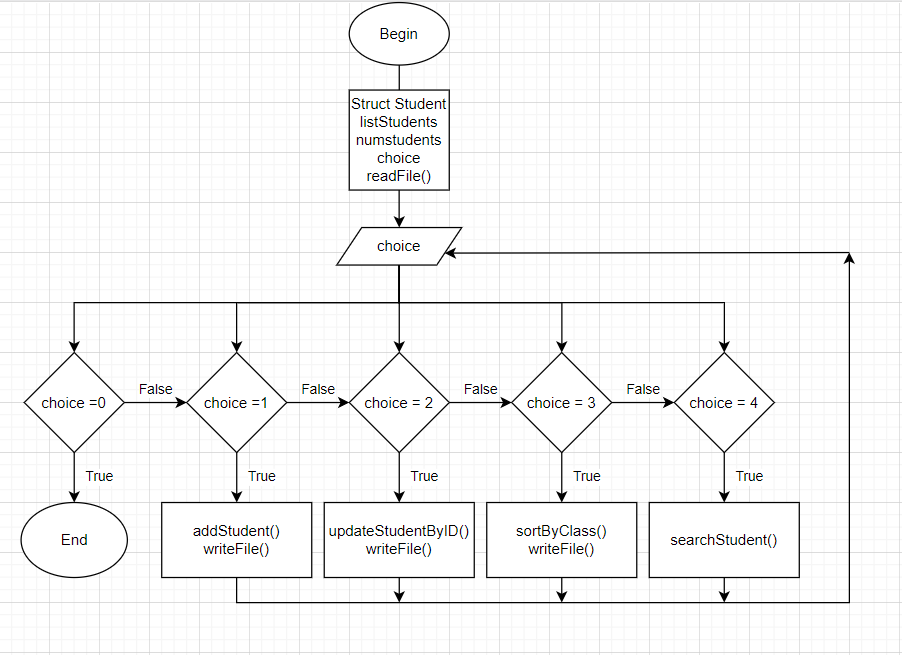
**Hàm viewListStudents():**

Đối với hàm này thì khá đơn giản , chỉ cần dùng vòng lặp for để in ra tất cả dữ liệu từ file.

**Hàm sortByClass() :**

Đầu tiên, ta dùng 2 vòng lặp for lồng nhau i, j để kiểm tra toàn bộ danh sách sinh viên và sử dụng câu lệnh if để kiểm tra, nếu vị trí đằng sau lớn hơn đằng trước thì từ đó thực hiện thuật toán đổi chỗ bằng việc khai báo struct Student với biến tempSV và thực hiện hai lần gán giá trị.

**Hàm main() :**

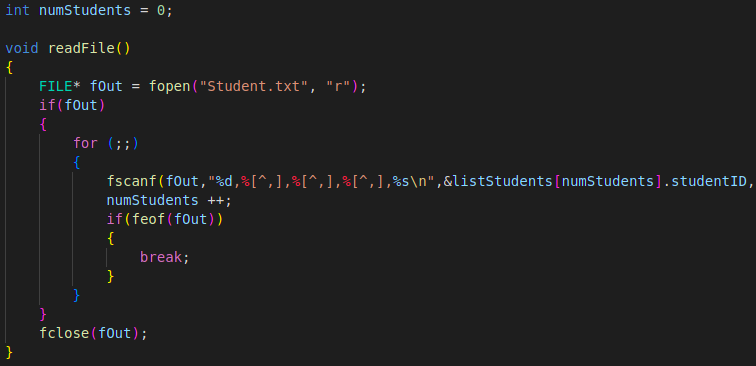
Để chương trình chạy đáp ứng được các chức năng thì ta sử dụng kết hợp câu lệnh điều kiện “ do – while” và hàm “ switch – case” với 4 case bao gồm: addStudent(), updateStudentByID(), sortByClass() và searchStudent() .

**2. Kỹ thuật sử dụng**

**\* Biến đếm**

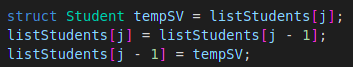
- Biến đếm là một trong các kỹ thuật căn bản, giúp ta kiểm soát số lượng các trường hợp quan trọng.

- Ta sẽ tạo một biến đếm với giá trị ban đầu là 0. Với mỗi lần trường hợp quan trọng xuất hiện, ta sẽ tăng biến đếm lên 1. Ví dụ như ở đây, em muốn đếm trong file có bao nhiêu sinh viên.

**\* Biến tạm**

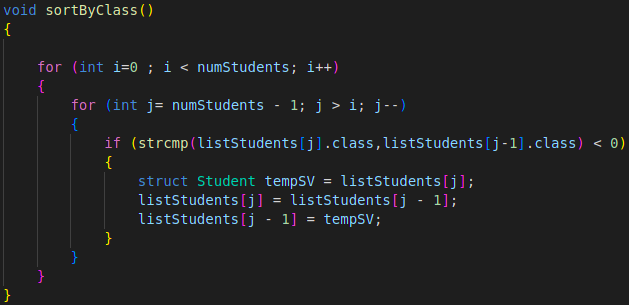
- Biến tạm là một kỹ thuật đơn giản dùng để lưu một giá trị của một biến khác trước khi nó bị thay đổi hoặc dùng để tính giá trị tạm thời cần được sử dụng ngay.

- Ví dụ ta có biến listStudents[j] và biến listStudents[j-1]. Ta cần hoán đổi giá trị của 2 biến listStudents[j] và listStudents[j-1]. Nhưng khi listStudents[j] = listStudents[j-1] thì lúc đó cả listStudents[j] và listStudents[j-1] đều có cùng một giá trị. Ta đã mất đi một giá trị còn lại nên ta cần một biến tạm để lưu giá trị của listStudents[j].

**\* Sắp xếp**

- Phương thức sắp xếp mảng đơn giản nhất đó là sử dụng giải thuật Interchange Sort. Giải thuật sử dụng 2 vòng lặp, với mỗi vòng lặp, lấy giá trị này xét với các giá trị khác trong mảng, nếu giá trị nào nhỏ hơn thì hoán đổi giá trị đó và giá trị đang xét.

- Ví dụ:



**3. Kiểm thử chương trình**

**CHƯƠNG 3: HƯỚNG PHÁT TRIỂN**

**1. Hạn chế**

Bên cạnh những kết quả đạt được, chương trình vẫn còn một số hạn chế sau:

- Chương trình mới chỉ hỗ trợ quản lý hồ sơ sinh viên, còn quản lý nhân viên, quản lý nộp - rút hồ sơ và thống kê báo cáo thì hệ thống chưa hỗ trợ.

- Chương trình chỉ dành cho nhân viên của trường sử dụng, sinh viên không có quyền sử dụng hệ thống để có thể tìm kiếm và xem thông tin của hồ sơ.

**2. Hướng phát triển**

Trong tương lai để hoàn thiện và hoạt động hiệu quả hơn, dự án đề ra những định hướng phát triển sau:

- Thêm chức năng đăng nhập vào hệ thống để dễ dàng xác minh người dùng

- Thêm xử lý dữ liệu nội bộ trong trường học

- Khảo sát và cải tiến các chức năng cho phù hợp với nghiệp vụ quản lý hồ sơ sinh viên trong trường học

**CHƯƠNG 4: KẾT LUẬN**

Sau một thời gian nghiên cứu và lắp đặt, dự án đã đạt được những kết quả nhất định và hơn nữa bản thân em đã trau dồi và học hỏi được rất nhiều kiến thức quan trọng trước khi hoàn thành khóa học. Kết quả thu được như sau:

- Có kiến thức cơ bản về ngôn ngữ lập trình C sử dụng trong dự án như: struct, các hàm, mảng, đọc / ghi tệp, ...

- Trên cơ sở những kiến thức đã tiếp thu để vận dụng và xây dựng hệ thống với các chức năng cơ bản và đáp ứng một số yêu cầu của hệ thống quản lý hồ sơ sinh viên.

- Giao diện của hệ thống thân thiện, dễ sử dụng.