 **TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI: QUẢN LÝ NHÂN VIÊN**

**Giảng viên hướng dẫn: Trần Thị Dung**

**Thực hiện:**

**Huỳnh Xuân Lãm** **MSSV:6051071062**

**Bùi Minh Thiện MSSV:6051071113**

**Võ Quang Vinh MSSV:6051071143** **Lớp: CQ.60.CNTT Khóa: K60**

TP.Hồ Chí Minh năm 2020

**LỜI CẢM ƠN**

Trong thời đại công nghệ thông tin ngày nay, việc ứng dụng Công nghệ thông tin vào cuộc sống cũng như các ngành khoa học ngày càng trở nên quan trọng. Quá trình làm đề tài thực tập chuyên ngành này là bước đầu tiên đi vào thực tiễn và cũng chính là bước đầu thực hành và đúc rút chứng minh cho những môn học trên ghế nhà trường nói chung và môn học chuyên ngành nói riêng. Nó cũng chính là quá trình nhận xét đánh giá và rút ra ưu, nhược điểm, để từ đó tìm ra phương án tối ưu nhất cho việc học và đi ra thực tế sau này của bản thân.

Chúng em xin chân thành cảm ơn sự hướng dẫn của các thầy, cô giáo trong khoa Công nghệ thông tin – Trường Đại học Giao thông Vận tải phân hiệu tại TP. Hồ Chí Minh đã tạo mọi điều kiện và luôn giúp đỡ, hướng dẫn nhóm em tận tình để nhóm hoàn thành bài tập lớn chuyên ngành này. Chúng em chân thành cảm ơn các thầy cô luôn luôn sẵn sàng giúp đỡ và tạo mọi điều kiện tốt nhất cho nhóm em.

Mặc dù có nhiều cố gắng bằng toàn bộ kiến thức để hoàn thành công việc, song thời gian và kinh nghiệm của bản thân chưa được trau dồi nhiều nên việc trình bày, phân tích, xây dựng chương trình còn nhiều kiến thức cần được bổ sung. Vì vậy, chúng em rất mong nhận được ý kiến đóng góp của thầy cô để sản phẩm này có thể hoàn thiện, được ứng dụng vào thực tiễn.

**LỜI NÓI ĐẦU**

Ngày nay Công nghệ thông tin đã trở thành một phần tất yếu của cuộc sống con người, có người đã nói rằng nước Mỹ giàu mạnh một phần là nhờ và Công nghệ thông tin. Nếu lúc trước Công nghệ thông tin là một điều viễn tưởng thì giờ đây nó đã trở thành một phần không thể thiếu trong tất cả các lĩnh vực từ y học, kinh doanh đến cả giáo dục.

Đất nước ta đang ngày một phát triển, đang cố gắng hòa nhập và rút ngắn khoảng cách với thế giới, việc nước ta trở thành thành viên của WTO đã được các nhà kinh doanh chú ý đến và đâu tư ngày càng nhiều vào Việt Nam. Vì vậy để không bị quá lạc hậu, để rút ngắn khoảng cách với các nước, để đáp ứng yêu cầu tất yếu của các nhà đầu tư vào Việt Nam và để các nhà kinh doanh trong nước có đủ sức cạnh tranh với nước ngoài thì bắt buộc phải đầu tư cho Công nghệ thông tin mà ở đây chính xác là các phần mềm tin học dùng cho các công ty, bệnh viện v.v. Việc áp dụng các phần mềm tin học vào các lĩnh vực giúp nâng cao tính hiệu quả và chính xác của công việc, ngoài ra còn tiết kiệm thời gian và giảm bớt mệt nhọc cho con ngƣời khiến hệ thống công việc hoạt động nhịp nhàng hơn.

Nhu cầu thực tế của xã hội đòi hỏi con người phải luôn năng động và sáng tạo để tạo ra ngày càng nhiều sản phẩm cho xã hội. Các công ty luôn luôn phát triển, các hồ sơ tuyển vào sẽ nhiều lên vì vậy đòi hỏi phải cần rất nhiều những kho chứa hồ sơ để lưu trữ hồ sơ của nhân viên khi vào công ty. để giảm đi những bất cập trong công tác quản lý nhân sự và lương trong công ty, giải pháp hiệu quả nhất hiện nay là đầu tư công nghệ và thiết bị hiện đại, ứng dụng công nghệ thông tin, tự động hoá trong điều hành và quản lý nhân sự để nâng cao chất lượng phục vụ, đổi mới phương thức quản lý, tăng năng xuất hiệu quả. Đó là những nội dung cơ bản đề cập đến trong đề tài này. Hôm nay chúng em chọn đề tài tốt nghiệp là “PHẦN MỀM QUẢN LÝ NHÂN SỰ”. Ai cũng biết công ty nào cũng vậy muốn tồn tại và phát triển thì phải có nhân tố con người. Cùng với tốc độ phát triển mạnh mẽ của Công nghệ thông tin các lập trình viên đã phát minh ra nhiều phần mềm hữu ích nhằm phục vụ cho công việc của con người và phần mềm quản lý nhân sự cũng là một trong những vấn đề con người quan tâm nhiều nhất, nó giúp cho công tác nghiệp vụ của các công ty giảm thiểu tối đa những vất vả trong công việc giúp cho việc lưu trữ hồ sơ dễ dàng hơn, giảm thiểu diện tích các kho và thậm chí là không cần.

Đề tài PHẦN MỀM QUẢN LÍ NHÂN VIÊN là một đề tài không phải là mới, hiện tại trên thị trường cũng có rất nhiều các phần mềm của các công ty phần mềm có chức năng tương tự và chuyên nghiệp, nhưng vì mong muốn có thể tìm hiểu một phần kiến thức nhỏ nhoi trong một trong một sa mạc Công nghệ thông tin nên chúng em đã đi vào tìm hiểu về cơ cấu tổ chức và phát triển công ty để có thể viết ra một phần mềm quản lí nhân viên nhằm phục vụ cho lợi ích của công ty. Ứng dụng phần mềm của chúng em có thể chưa hoàn chỉnh và còn nhiều thiếu sót nhưng đó là sự cố gắng của chúng em trong bước đầu hòa nhập vào môi trường làm việc bên ngoài và tiếp xúc chập chững với Công nghệ thông tin, chúng em mong quý thầy cô trong khoa có thể tạo điều kiện giúp đỡ chúng em hoàn thành đề tài tốt hơn.

Chương trình của chúng em được lập trình bằng ngôn ngữ lập trình C. Phần mềm này giúp công ty quản lí hồ sơ , lương, chế độ chế độ của nhân viên giúp giảm tải bớt việc tiềm kiếm mất thời gian mà hiệu quả và tính chính xác cao.

Nhóm thực hiện:

**Huỳnh Xuân Lãm**

**Bùi Minh Thiện**

**Võ Quang Vinh**

**MỤC LỤC**

1. **LỜI CẢM ƠN**
2. **LỜI NÓI ĐẦU**

**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* 1.1.Lý do chọn đề tài................................................................................................................................1
  + 1.2.Mục đích của đề tài............................................................................................................................1
  + 1.3.Nội dung nguyên cứu.........................................................................................................................2
  + 1.4.Phạm vi ứng dụng...............................................................................................................................2

**CHƯƠNG 2: GIỚI THIỆU VỀ ĐỀ TÀI**

* 2.1.Khảo sát thực tế ................................................................................................................................2
* 2.2.Mục tiêu.............................................................................................................................................2
* 2.3.Chức năng và nhiệm vụ......................................................................................................................3
* 2.4.Đảm bảo.............................................................................................................................................3

**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT**

1. **Phân tích chức năng**...........................................................................................................

1.Chức năng hệ thống................................................................................................................4

2.Chức năng quản lí nhân viên...................................................................................................4

3.Chức năng quản lí chế độ........................................................................................................5

4.Chức năng quản lí tiền lương..................................................................................................5

5.Chức năng tra cứu...................................................................................................................6

6.Chức năng thống kê, báo cáo..................................................................................................6

1. **Phân tích lí thuyết**................................................................................................................

**1.Làm việc với tệp**................................................................................................................7

1.1. Tại sao chúng ta cần đến tệp (file)............................................................................................7

1. File văn bản – text files....................................................................................................7
2. File nhị phân – Binary file................................................................................................7

1.2. Các thao tác với files..................................................................................................................8

1.3.Thao tác đóng files.....................................................................................................................9

1. 4.Đọc ghi file văn bản trong C.......................................................................................................9

1.5.Đọc ghi file nhị phân trong C....................................................................................................11

* Ghi file nhị phân ....................................................................................................11
* Đọc file nhị phân................................................................................................................12

**2.Cấu trúc liên kết đơn**.....................................................................................................13

2.[1. Khai báo linked list](#_Toc10334) ....................................................................................................................14

2.[2. Tạo mới 1 Node](#_Toc22273) .........................................................................................................................15

2.[3. Thêm Node vào danh sách liên kết...........................................................................................16](#_Toc9824)

[a. Thêm vào](#_Toc24029) .............................................................................................................................16

[b. Thêm vào cuối](#_Toc24648) ......................................................................................................................16

1. [Thêm vào vị trí bất kì](#_Toc20823) .........................................................................................................17

2.4.Xóa node khỏi danh sách liên kết....................................................................................................18

[a. Xóa đầu](#_Toc2789) ..................................................................................................................................18

[b. Xóa cuối](#_Toc20751) ..................................................................................................................................19

[c. Xóa vị trí bất kì](#_Toc1955) ........................................................................................................................19

2.[5. Lấy giá trị ở vị trí bất kỳ](#_Toc28381) ................................................................................................................20

2.[6. Tìm kiếm trong danh sách liên kết](#_Toc26049) ...............................................................................................21

2.7. Danh sách liên kết.........................................................................................................................21

**3. Các thuật toán sắp xếp, tìm kiếm**...................................................................

3.1. Thuật toán sắp xếp nỗi bọt..........................................................................................................22

3.2. Thuật toán sắp xếp chọn..............................................................................................................24

3.3. Thuật toán sắp xếp chèn..............................................................................................................27

3.4. Tìm kiếm tuyến tính.....................................................................................................................28

3.5. Tìm kiếm nhị phân........................................................................................................................39

1. **Xây dựng chương trình code**.......................................................................................30

**CHƯƠNG 4: HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH**....................................................41

**CHƯƠNG 5:** **TỔNG KẾT**..........................................................................................................................42

**1.Kết quả đạt được**

**2. Định hướng phát triển**

**3. Tài liệu tham khảo**

**IV**. **KẾT LUẬN**.................................................................................................................................................44

**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

**1.1.Lý do chọn đề tài**

Quản lí nhân sự là một trong những bộ phận quan trọng trong công ty, đặc biệt là trong các công ty lớn trong nước và ngoài nước. Sự thành bại của công ty phụ thuộc vào cách thức tổ chức nhân sự có tốt hay không. Trong năm vừa qua quản lí nhân viên đang dần phát triển mạnh mẽ không những ở các công ty nước ngoài mà các công ty tư nhân cũng đã dần nhận thấy sự quan trọng của cơ cấu tổ chức nhân sự trong công ty.

Mỗi lần muốn tìm hồ sơ của một nhân viên nào đó trong công ty người quản lí nhân viên lại phải tìm lần lượt trong kho chứa xem hồ sơ nhân viên đó nằm ở đâu. Như vậy mất rất nhiều thời gian mà có khi không tìm ra do hồ sơ nhân viên quá nhiều. Công việc lưu trữ hồ sơ rất thủ công, hầu hết ở các khâu làm cho số lượng nhân viên tham gia vào việc lưu trữ hồ sơ sẽ nhiều lên. Việc thống kê báo cáo hàng tháng, hàng năm sẽ gặp rất nhiều khó khăn nếu thông tin nhân viên ngày càng nhiều.

Khi có nhân viên trong công ty thôi việc người quản lí nhân viên công ty phải tìm trong kho xem hồ sơ đó ở đâu để chuyển sang kho lưu trữ thôi việc, nếu hàng ngày cứu lặp đi lặp lại nhiều lần như vậy người quản lí nhân sự sữ gặp rất nhiều khó khăn trong công tác lưu trữ và quản lí.

**1.2.Mục đích của đề tài**

Quản lí nhân viên là một trong những vấn đề then chốt trong mọi công ty. Quản lí nhân sự không tốt sẽ dẫn đến nhiều vấn đề bất cập trong công ty. Chúng ta luôn thấy những tình trạng chấm nhầm công hoặc lưu trữ sai thông tin nhân viên, như vậy sẽ dẫn đến những xáo trộn khi phát lương...Vì vậy, yêu cầu đặt ra ở đây là làm thế nào để có thể lưu trữ hồ sơ và lương của nhân viên một cách hiệu quả nhất, giúp cho nhân viên trong công ty có thể yên tâm và thoải mái làm việc và đạt kết quả cao.

Dựa trên nhu cầu thực tế của xã hội nhằm giải quyết để giảm đi những bất cập trong công tác quản lí nhân sự và lương trong công ty, giải pháp hiệu quả nhất hiện nay là đầu tư công nghệ và thiết bị hiện đại, ứng dụng công nghệ thông tin và tự động hóa trong điều hành và quản lí nhân sự để nâng cao chát lượng phục vụ, đổi mới phương thức quản lí, tăng năng suất hiệu quả ...Đó là những nội dung cơ bản đề cập đến trong đề tài này.

Phần mềm quản lí nhân viên sẽ là người hỗ trợ đắc lực cho việc quản lí nhân viên của công ty. Bạn không cần thiết phải biết nhiều về tin học, bạn vẫn có thể sử dụng phần mềm quản lí nhân viên một cách hiệu quả. Giao diện thuận tiện dễ sử dụng của chương trình sẽ luôn khiến bạn cảm thấy thật thoải mái khi sử dụng phần mềm này.

**1.3.Nội dung nguyên cứu**

Để có thể hoàn thành được đề tài thì cần phải nguyên cứu những nội dung sau :

* Tìm hiểu tổ chức, cách thức hoạt động của công ty.
* Tìm hiểu về các nghiệp vụ của công ty như nghiệp vụ tính lương, nghiệp vụ chấm công, nghiệp vụ quản lí bảo hiểm xã hội , phải nắm rõ những nghiệp vụ này chúng ta mới có thể xây dựng được phần mềm.
* Cần phải tìm hiểu nguyên cứu về những nhu cầu có thể phát sinh của doanh nghiệp trong tương lai để từ đó có thể có giải pháp thiết kế phần mềm phù hợp có tính mềm dẻo cao đáp ứng được bất cứ sự thay đổi có thể xảy ra nào.

**1.4.Phạm vi ứng dụng**

Ứng dụng được xây dựng với mục đích là phục vụ cho tác vụ quản lí hồ sơ nhân viên, quản lí lương, chấm công, cũng như là những tác vụ luân chuyển nhân viên, quản lí bảo hiểm, các chế độ.

Vì vậy đối tượng sử dụng nó là những nhân viên có thẩm quuền của doanh nghiệp, cụ thể hơn đó chính là những nhà lãnh đạo doanh nghiệp và những nhân viên thuộc phòng nhân sự của công ty.

**CHƯƠNG 2: GIỚI THIỆU VỀ ĐỀ TÀI**

**2.1.Khảo sát thực tế**

Quản lí nhân viên là một trong những công việc vô cùng quan trọng, góp phần quyết định sự thành bại của một công ty, do nhu cầu thực tiễn như vậy.

Phần mềm quản lí của nhân viên để phục vụ công tác quản lí nhân viên, người quản lí có khả năng bao quát và điều hành nhân sự, chịu trách nhiệm đưa ra các thống kê, báo cáo định kì và thường xuyên về tình hình hoạt động của công ty. Các khâu chuyên trách khác nhau của nhân sự do nhân viên trong bộ phận nhân sự quản lí đảm nhiệm.

**2.2.Mục tiêu**

Nguyên cứu bài toán quản lí nhân viên để thiết kế và lập trình phần mềm nhằm hỗ trợ công tác quán lí của công ty.

1. ***Cập nhật hồ sơ nhân viên*** mới vào công ty.
2. Quản lí chế độ cho nhân viên giúp cho nhân viên được hưởng các chế độ ưu đãi cũng như kỉ luật của công ty.
3. ***Tìm kiếm, tra cứu thông tin*** nhân viên trong công ty.
4. ***Quản lí luân chuyển nhân viên ...***

**2.3.Chức năng và nhiệm vụ**

**Một vài chức năng giúp tối ưu hóa việc quản lý nhân sự của**

* Hệ thống yêu cầu đăng nhập trước khi quản trị (hay nhân viên quản lí nhân sự) muốn quản lí và sử dụng các chức năng quản lí tương ứng với quyền hạn của người đăng nhập( quản lí người dùng).
* Menu chính bao gồm các menu con, mỗi menu thường là điểm vào của một chức năng.
* Chức năng *hệ thống*( cập nhật các thông tin nhân viên (thêm, xóa, sửa) chỉ được sử dụng bởi người quản lí.
* Chức năng tìm kiếm có thể áp dụng cho : quản lí (bộ phận nhân viên phòng nhân sự).
* Chức năng *quản lí chế độ* chỉ được sử dụng bởi người quản trị phần mềm này thuộc phần quyền hạn.
* *Kết xuất báo cáo, thống kê* chỉ được áp dụng cho nhóm người quản lí thuộc phòng nhân sự.
* *Chức năng quản lí lương* chỉ được áp dụng cho người trực tiếp quản lí bộ phận làm lương cho nhân viên trong công ty thông qua một số thông tin như bảng chấm công, danh sách làm thêm giờ ...
* Chức năng *thông báo* danh sách nhân viên thôi việc của công ty khi có yêu cầu của cấp trên chỉ được thực hiện bởi nhân viên phòng nhân sự.

**2.4.Đảm bảo**

Hệ thống quản lí ***quản lí nhân sự được thiết kế*** đảm bảo:

* Thời gian xử lí nhanh.
* Tra cứu dữ liệu được xử lí chính xác các thông tin nhân viên trong công ty nhằm phục vụ cho các thao tác hoặc các chức năng sau nó.
* Dễ sử dụng với những giao diện đồ họa thân thiện phù hợp với người sử dụng.

**CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG**

1. **Phân tích chức năng**
2. **Chức năng hệ thống**

Hệ thống yếu cầu đăng nhập trước khi quản trị (hay nhân viên quản lí nhân sự ) muốn quản lí , và sử dụng các chức năng quản lí tương ứng với quyền hạn của người đăng nhập (quản lí, người dùng).

Ở đây người người quản trị có thể thay đổi mật khẩu của chương trình để đảm bảo tính bảo mật của chương trình.

**Hệ thống**

Sơ đồ 3.1.a

đồ 3.1.a

1. **Chức năng quản lí nhân viên**

Chức năng “quản lí nhân viên” ở chức năng này người trực tiếp quản lí đó là nhân viên phòng nhân sự thực hiện được những yêu cầu sau:

+ Cập nhật tất cả các thông tin của nhân viên trong công ty như (Họ tên, năm sinh, quê quán,...) bổ sung vào danh sách nhân viên trong công ty.

+ Nhập bộ phận của từng nhân viên vào phòng ban cụ thể nào đó

+ Sửa chi tiết thông tin, nhân viên: Thực hiện việc chỉnh sửa, cập nhật bổ sung thêm các thông tin chi tiết của từng nhân viên.

+ Lập phân loại, thực hiện việc cập nhật bổ sung, chỉnh sửa phân loại nhân viên vào từng phòng ban cụ thể

+ Xóa thông tin: Thực hiện việc xóa thông tin khi nhân viên nghỉ việc hoặc chuyển đi nơi khác..

+ Báo cáo cấp trên.

Quản lí nhân sự

Sơ đồ 3.1.b

Sơ đồ 3.1.b

Bộ phận phòng ban

Thông tin nhân viên

1. **Chức năng quản lí chế độ**

Chức năng này sẽ quản lí những cán bộ công nhân viên trong công ty có tham gia các chế độ như: Bảo hiểm, chế độ khen thưởng, nhân viên kỉ luật, chế độ thai sản..và sẽ thống kê lại trong từng tháng.

+ Mở cơ sở dữ liệu.

+ Cho phép người quản lí thêm thông tin mới, chỉnh sửa các thông tin và xóa bỏ các thông tin không cần thiết khi nhân viên đó chuyển đi. Thông qua chức năng này giúp cho việc quản lí dễ dàng hơn.

+ Thống kê lại từng tháng báo cáo cấp trên. Sơ đồ 3.1.c

Sơ đồ 3.1.c Sơ đồ 3.1.c

Chế độ thai sản

Nhân viên kỉ luật

Bảo hiểm

Chế độ khen thưởng

Quản lí chế độ

1. **Chức năng quản lí tiền lương**

Chức năng này sẽ được nhân viên phòng nhân sự theo dõi quá trình làm việc và tính lương cho từng cán bộ công nhân viên trong từng tháng như: làm thêm giờ của công nhân, bảng chấm công , bảng lương của công nhân viên trong công ty...

+ Theo dõi chấm công cho nhân viên từng ngày.

+ Cập nhật xem có nhân viên nào được khen thưởng hay bị kỉ luật.

+ Tình hình công việc cảu mỗi nhân viên xem có tăng ca thêm hay giảm giờ làm không.

+ Chức năng này sẽ được làm hàng ngày , phục vụ cho việc tính lương hàng tháng của mỗi nhân viên.

Quản lí tiền lương

Sơ đồ 3.1.d

Bảng lương

Bảng công

Làm thêm giờ

1. **Chức năng tra cứu**

Chức năng này dùng tra cứu thông tin mà người quản trị muốn tìm kiếm như tra cứu thông tin của một nhân viên nào đó chẳng hạn, xem nhân viên như thế nào, theo dõi tình hình làm việc của nhân viên đó...

Chức vụ

Tình hình làm việc

Thông tin cá nhân

Tra cứu

Sơ đồ 3.1.e

Thực hiện được tất cả những công việc có khi yêu cầu cảu cấp trên muốn biết thông tin nào đó

**Ví dụ:** Muốn biết tình hình làm việc của nhân viên A thuộc phòng ban nào đó, hay chức vụ gì...

1. **Chức năng thống kê báo cáo**

Báo cáo theo định kì hoặc theo yêu cầu của cấp trên các thông tin của nhân viên như : Bảng công, khen thưởng, chế độ thai sản hợp đồng, bảo hiểm, thôi việc...

Hợp đồng, thôi việc ...

Bảo hiểm, chế độ thai sản

Khen thưởng, kỉ luật

Bảng công

Thống kê báo cáo

Sơ đồ 3.1.f

1. **Phân tích lí thuyết**

**1. Làm việc với tệp**

* 1. **. Tại sao sao chúng ta cần đến tệp (file)**

Dữ liệu được lưu ở biến của chương trình, và nó sẽ biến mất khi chương trình kết thúc. Sử dụng file để lưu trữ dữ liệu cần thiết để đảm bảo dữ liệu của chúng ta không bị mất ngay cả khi chương trình của chúng ta ngừng chạy.

Nếu chương trình của ta có đầu vào(input) là lớn, ta sẽ rất vất vả nếu phải nhập mỗi ngày khi chạy.

Thay vào đó, hãy lưu file của chương trình của mình sẽ tự đọc mỗi lần khi chạy.

Dễ dàng sao chép, di chuyển dữ liệu giữa các thiết bị với nhau.

1. **[File văn bản – text files](https://nguyenvanhieu.vn/doc-ghi-file-trong-c/" \l "1-file-van-ban-%e2%80%93-text-files" \o "1. File văn bản – text files)**

File văn bản là file thường có đuôi là .txt. Những file này bạn có thể dễ dàng tạo ra bằng cách dùng các text editer thông dụng như Notepad, Notepad++, Sublime Text,…

Khi chúng ta mở các file này bằng các text editer nói trên, ta sẽ thấy được văn bản ngay và có thể dễ dàng thao tác sửa, xóa, thêm nội dung của file này.

* Kiểu file này thuận tiện cho chúng ta trong việc sử dụng hàng ngày, nhưng nó sẽ kém bảo mật và cần nhiều bộ nhớ để lưu trữ hơn.

**b.**[**File nhị phân – Binary files**](https://nguyenvanhieu.vn/doc-ghi-file-trong-c/#2-file-nhi-phan-%e2%80%93-binary-files)

File nhị phân thường có đuôi mở rộng là **.bin**

Thay vì lưu trữ dưới dạng văn bản thuần thúy, các file này được lưu dưới dạng nhị phân, chỉ bao gồm các số 0 và 1. Ta cũng sẽ thấy các con số này nếu cố mở nó bằng 1 text editer kể trên.

Loại file này giúp lưu trữ được dữ liệu với kích thước lớn hơn, không thể đọc bằng các text editer thông thường và thông tin lưu trữ ở loại file được bảo mật hơn so với file văn bản.

**1.2. Các thao tác với file**

Khi làm việc với file, ta cần khai báo 1 con trỏ kiểu FILE. Việc khai báo này là cần thiết để có sự kết nối giữa chương trình của mình và tập tin mà mình cần thao tác.

FILE \*fptr;

Để đọc ghi file trong C cũng như trong mọi ngôn ngữ lập trình, việc đầu tiên mình cần làm là mở file mà mình muốn làm việc. Trong ngôn ngữ lập trình C, chúng ta có thể mở file bằng cách sử dụng hàm  fopen() trong thư viện stdio.h như sau:

fptr = fopen("fileopen","mode");

|  |  |  |
| --- | --- | --- |
| Mode | Ý nghĩa | Nếu file không tồn tại |
| r | Mở file chỉ cho phép đọc | Nếu file không tồn tại, fopen() trả về NULL. |
| rb | Mở file chỉ cho phép đọc dưới dạng nhị phân. | Nếu file không tồn tại, fopen() trả về NULL. |
| w | Mở file chỉ cho phép ghi. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| Wb | Mở để viết chế độ in nhị phân | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| a | Mở file ở chế độ ghi “append”. Tức là sẽ ghi vào cuối của nội dung đã có. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| ab | Mở file ở chế độ ghi nhị phân “append”. Tức là sẽ ghi vào cuối của nội dung đã có. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| r+ | Mở file cho phép cả đọc và ghi. | Nếu file không tồn tại, fopen() trả về NULL. |
| rb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. | Nếu file không tồn tại, fopen() trả về NULL. |
| w+ | Mở file cho phép cả đọc và ghi. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| wb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. | Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động. |
| a+ | Mở file cho phép đọc và ghi “append”. | Nếu file không tồn tại, nó sẽ được tạo tự động. |
| ab+ | Mở file cho phép đọc và ghi “append” ở dạng nhị phân. | Nếu file không tồn tại, nó sẽ được tạo tự động. |

Trong đó mode là một tham số chúng ta cần chỉ định.

**Bảng 3.2**

.

*Bảng Mode*

**1.3. Thao tác đóng file**

Khi làm việc với tập tin hoàn tất, kể cả là file nhị phân hay file văn bản. Ta cần đóng file sau khi làm việc với nó xong.

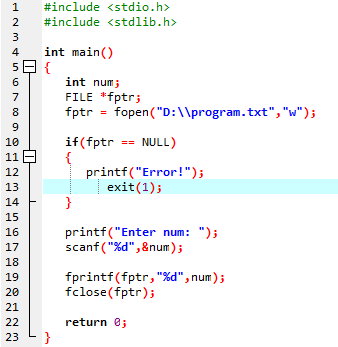
Việc đóng file đang mở có thể được thực hiện bằng cách dùng hàm fclose().

**1.4. Đọc/Ghi file văn bản trong C**

Chúng ta sẽ học cách đọc ghi file trong C với file văn bản trước. Với file nhị phân, bạn kéo xuống dưới để xem tiếp.

Để làm việc với file văn bản, chúng ta sẽ sử dụng fprintf() và fscanf().

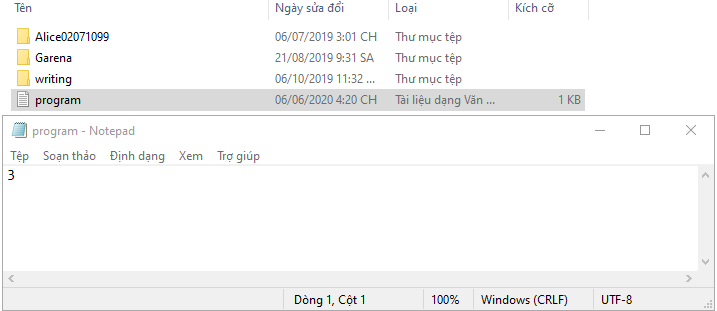
VD1: Ghi file dùng fprintf().

****

Hình 4.1

Chương trình nhận số num từ bàn phím và ghi vào file văn bản program.txt.

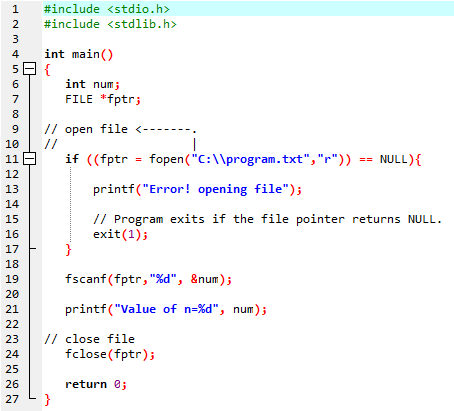
Sau khi chạy chương trình này, sẽ thấy file văn bản program.txt được tạo mới trong ổ C trên máy tính của chúng ta. Khi mở file này lên, sẽ thấy số mà mình vừa nhập cho biến num kia.

****

Hình 4.2

Chương trình này sẽ đọc giá trị số được lưu trong file program.txt mà chương trình ở **VD1** vừa tạo ra và in lên màn hình

**VD2:** Đọc sử dụng fscanf().



Hình 4.3

Chương trình này sẽ đọc giá trị số được lưu trong file program.txt mà chương trình ở VD1 vừa tạo ra và in lên màn hình.

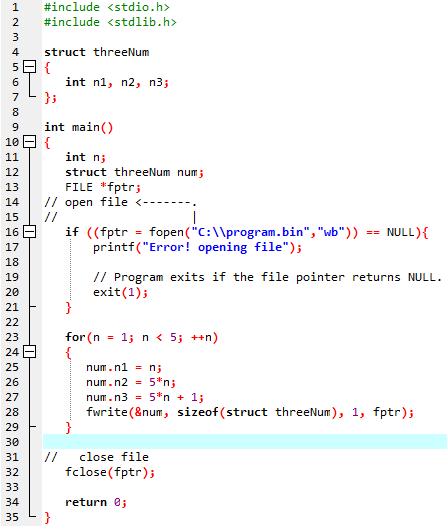
**1.5. Đọc/ghi file nhị phân trong C**

Các hàm fread() và fwrite() trong C được sử dụng để đọc và ghi file trong C ở dạng nhị phân.

\* **Ghi file nhị phân**

Để ghi file nhị phân, bạn cần sử dụng hàm fwrite(). Hàm này cần 4 tham số: địa chỉ của biến lưu dữ liệu cần ghi, kích thước của biến lưu dữ liệu đó, số lượng kiểu dữ liệu của biến đó và con trỏ FILE trỏ tới file mà chúng ta muốn ghi. fwrite(address\_data,size\_data,numbers\_data,pointer\_to\_file);

**VD3:** Ghi file nhị phân



Hình 5a

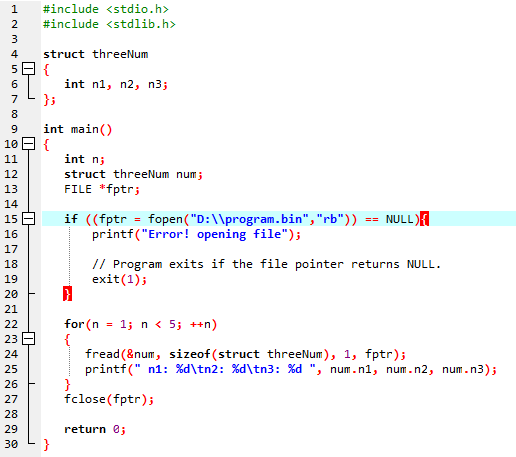
Trong **VD3** này, chương trình sẽ tạo ra một file program.bin trên ổ đĩa C của ta. Chương trình này đã khai báo 1 kiểu dữ liệu cấu trúc lưu 3 giá trị số n1, n2, n3; Và nó được sử dụng trong hàm main có tên biến là num.

#### \* Đọc file nhị phân

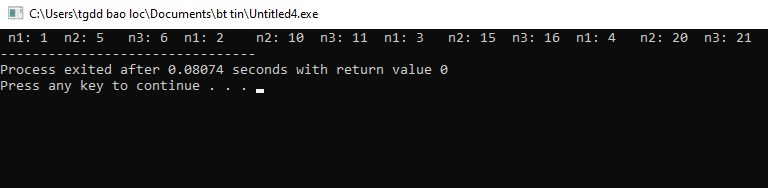
Hàm fread() cũng có 4 tham số tương tự như hàm fwrite() phía trên.

fread(address\_data,size\_data,numbers\_data,pointer\_to\_file);

VD 4: Đọc file nhị phân



Hình 5b

Trong **VD4** này, bạn đọc file program.bin và lặp qua từng dòng. Bạn sẽ nhận được các giá trị tương ứng khi bạn ghi vào trong VD3.

Hình 5c

**2. Cấu trúc liên kết đơn**

Về bản chất, danh sách liên kết có chức năng như một mảng, có thể thêm và xóa các phần tử ở bất kỳ vị trí nào khi cần thiết. Một số sự khác nhau giữa danh sách liên kết và mảng:

**Bảng 3.2.2**



**Bảng so: sánh mảng và danh sách liên kết**

**Lưu ý:** Ở bảng phía trên, các phần in nghiêng thể hiện đó là ưu điểm so với cột còn lại.

Danh sách liên kết đơn là một tập hợp các Node được phân bố động, được sắp xếp theo cách sao cho mỗi Node chứa “một giá trị”(Data) và “một con trỏ”(Next). Con trỏ sẽ trỏ đến phần tử kế tiếp của danh sách liên kết đó. Nếu con trỏ mà trỏ tới NULL, nghĩa là đó là phần tử cuối cùng của linked list.

Hình ảnh mô tả cho một Node trong danh sách liên kết đơn:



Hình 3.2.2a

Và đây là hình ảnh mô phỏng một danh sách liên đơn kết đầy đủ:

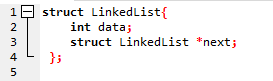


Hình 3.2.2b

*Cài đặt danh sách liên kết đơn:*

* 1. **Khai báo linked list**

Để đơn giản hóa, data của chúng ta sẽ là số nguyên(int). Chúng ta cũng có thể sử dụng các kiểu nguyên thủy khác(float, char,…) hay kiểu dữ liệu struct(SinhVien, CanBo,…) tự tạo.

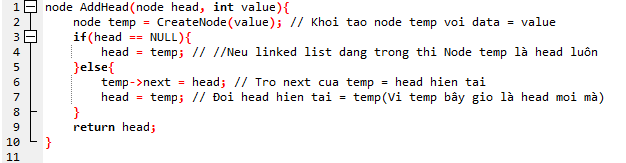


Hình 1.1

Khai báo trên sẽ được sử dụng cho mọi Node trong linked list. Trường data sẽ lưu giữa giá trị và next sẽ là con trỏ để trỏ đến kế tiếp của nó.

**Tại sao next lại là kiểu LinkedList của chính nó?**

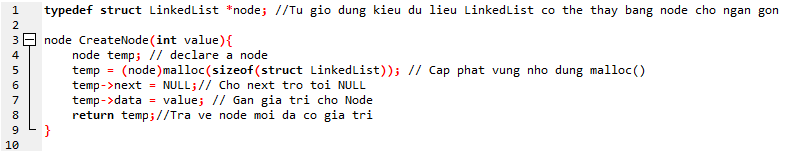
Bởi vì nó là con trỏ trỏ của chính bản thân nó, và nó trỏ tới một Node kế tiếp cũng có kiểu LinkedList.



### Hình 1.2

### Tạo mới 1 Node

Hãy tạo một kiểu dữ liệu của struct LinkedList để code thuận lợi hơn:



Hình 2.1

Mỗi một Node khi được khởi tạo, chúng ta cần cấp phát bộ nhớ cho nó, và mặc định cho con trỏ next trỏ tới NULL. Giá trị của Node sẽ được cung cấp khi thêm Node vào linked list.

* **typedef** được dùng để định nghĩa một kiểu dữ liệu trong C. VD: typeder long long LL;
* **malloc** là hàm cấp phát bộ nhớ của C. Với C++ chúng ta dùng new
* **sizeof** là hàm trả về kích thước của kiểu dữ liệu, dùng làm tham số cho hàm malloc

**Lưu ý:** Không giống với mảng, cần khai báo arr[size]. Trong linked list, vì mỗi Node sẽ có con trỏ liên kết đến Node tiếp theo. Do đó, với danh sách liên kết đơn, chỉ cần lưu giữ Node đầu tiên(HEAD). Có head rồi ta có thể đi tới bất cứ Node nào.

* 1. **Thêm Node vào danh sách liên kết**

1. **Thêm vào**

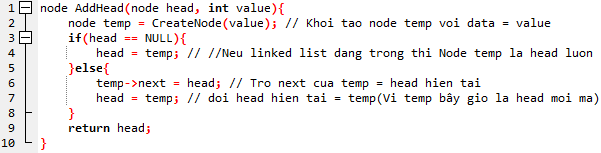
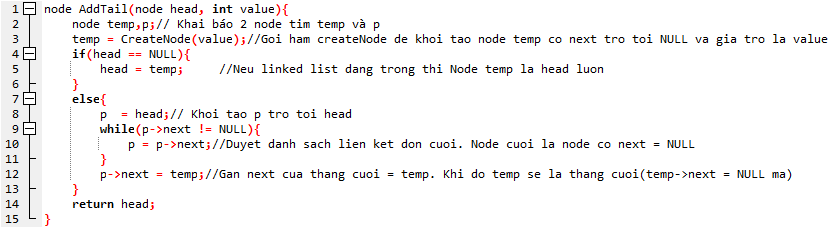
Việc thêm vào đầu chính là việc cập nhật lại head. Ta gọi Node mới(temp), ta có:

* Nếu head đang trỏ tới NULL, nghĩa là linked list đang trống, Node mới thêm vào sẽ làm head luôn
* Ngược lại, ta phải thay thế head cũ bằng head mới. Việc này phải làm theo thứ tự như sau:
  + Cho next của temp trỏ tới head hiện hành
  + Đặt temp làm head mới

#### Thêm vào cuối

Chúng ta sẽ cần Node đầu tiên, và giá trị muốn thêm. Khi đó, ta sẽ:

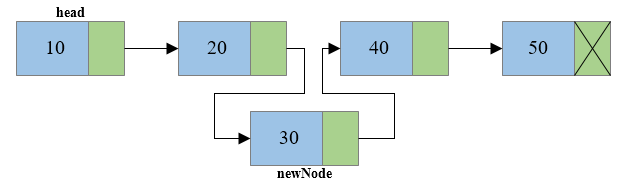
1. Tạo một Node mới với giá trị value
2. Nếu head = NULL, tức là danh sách liên kết đang trống. Khi đó Node mới(temp) sẽ là head luôn.
3. Ngược lại, ta sẽ duyệt tới Node cuối cùng(Node có next = NULL), và trỏ next của cuối tới Node mới(temp).



Hình 3.1

Tổng quan hơn, chúng ta sẽ sẽ viết hàm thêm một Node vào vị trí bất kỳ.

1. **Thêm vào vị trí bất kì**



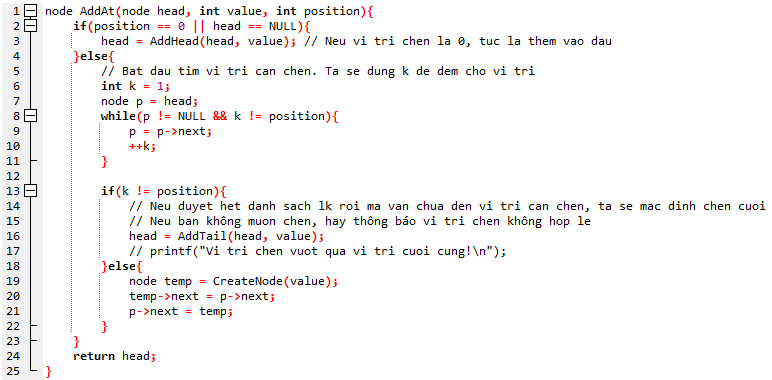
Hình 3.2

*Thêm Node vào giữa danh sách liên kết:*

Để làm được việc này, ta phải duyệt từ đầu để tìm tới vị trí của Node cần chèn, giả sử là Node Q, khi đó ta cần làm theo thứ tự sau:

* Cho next của Node mới trỏ tới Node mà Q đang trỏ tới
* Cho Node Q trỏ tới Node mới

Lưu ý: Chỉ số chèn bắt đầu từ chỉ số 0 .



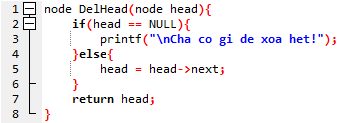
Hình 3.3

Lưu ý: Chúng ta phải làm theo thứ tự trên, nếu mình cho p->next = temp trước. Khi đó, mình sẽ không thể lấy lại phần sau của danh sách liên kết nữa(Vì next chỉ được được lưu trong p->next mà thay đổi p->next rồi thì còn đâu giá trị cũ).

* 1. **Xóa Node khỏi danh sách liên kết**

#### Xóa đầu

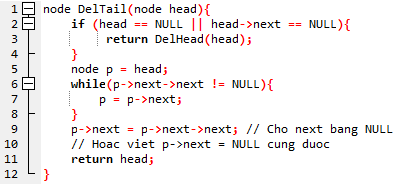
Chỉ cần cho biến kế tiếp của head làm head là được. Mà biến kế tiếp của head chính là head->next.



Hình 4.1

1. **Xóa cuối**

Xóa cuối thì phải duyệt đến cuối – 1, cho next của cuối – 1 đó bằng NULL.

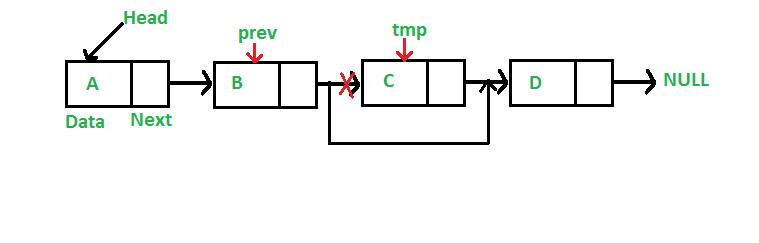


Hình 4.2

Node cuối – 1 là có p->next->next = NULL. Chúng ta cho next của nó bằng NULL là xong.

1. **Xóa vị trí bất kì**

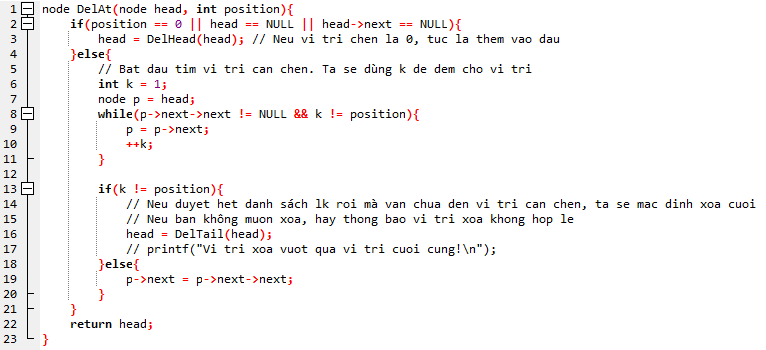
Node cuối – 1 là có p->next->next = NULL. Chúng ta cho next của nó bằng NULL là xong.



*Hình 4.3*

*Xóa node trong danh sách liên kết*

Lưu ý: Chỉ số xóa bắt đầu từ 0. Việc tìm vị trí cần xóa chỉ duyệt tới Node gần cuối thôi(cuối – 1). Sau đây là code xóa Node ở vị trí bất kỳ.

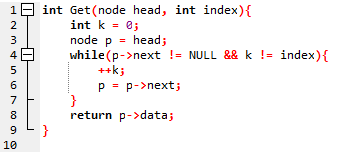


### Hình 4.4

### 

### Lấy giá trị ở vị trí bất kỳ

Chúng ta sẽ viết một hàm để truy xuất giá trị ở chỉ số bất kỳ. Trong trường hợp chỉ số vượt quá chiều dài của linked list – 1, hàm này trả về vị trí cuối cùng. Do hạn chế là chúng ta không thể raise error khi chỉ số không hợp lệ. Ta mặc định chỉ số mình truyền vào phải là số nguyên không âm. Nếu chúng ta muốn kiểm tra chỉ số hợp lệ thì nên kiểm tra trước khi gọi hàm này.



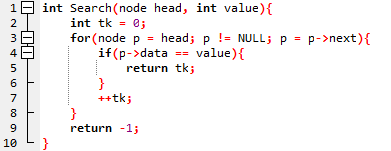
Hình 5.1

Lý do dùng p->next != NULL là vì chúng ta chỉ muốn đi qua các phần tử có value.

* 1. **Tìm kiếm trong danh sách liên kết**

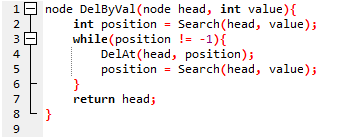
Hàm tìm kiếm này sẽ trả về chỉ số của Node đầu tiên có giá trị bằng với giá trị cần tìm. Nếu không tìm thấy, chúng ta trả về -1.

**VD 1**: tìm kiếm danh sách liên kết đơn



Hình 6.1

Chúng ta có thể sử dụng hàm này để xóa tất cả các Node trong danh sách liên kết có giá trị chỉ định như sau:

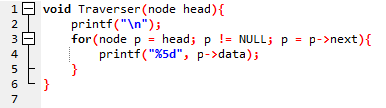


Hình 6.2

### Duyệt danh sách liên kết

Việc duyệt danh sách liên kết cực đơn giản. Khởi tạo từ Node head, chúng ta cứ thế đi theo con trỏ next cho tới trước khi Node đó NULL.

**VD 1**: duyệt danh sách liên kết



Hình 7.1

**3. Các thuật toán sắp xếp, tìm kiếm**

**3.1. Thuật toán sắp xếp nổi bọt**

**Ý tưởng thuật toán sắp xếp nổi bọt:**

Thuật toán sắp xếp nổi bọt thực hiện sắp xếp dãy số bằng cách lặp lại công việc đổi chỗ 2 số liên tiếp nhau nếu chúng đứng sai thứ tự (số sau bé hơn số trước với trường hợp sắp xếp tăng dần) cho đến khi dãy số được sắp xếp.

**Ví dụ minh họa:**

Giả sử chúng ta cần sắp xếp dãy số [ 5 1 4 2 8 ] này tăng dần.

**Lần lặp đầu tiên:**

( **5 1** 4 2 8 ) -> ( **1 5** 4 2 8 ), Ở đây, thuật toán sẽ so sánh hai phần tử đầu tiên và đổi chỗ cho nhau do 5 > 1.

#include <stdio.h>

void traodoi(int &x, int &y){

int temp = x;

x = y;

y = temp;

}

// hàm sap xep noi bot

void SapXepNoiBot(int a[], int n){

int i, j;

bool cotraodoi = false;

for (i = 0; i < n - 1; i++){

//i phan tu cuoi cung da duoc sap xep cotraodoi = false;

for (j = 0; j < n - i - 1; j++){

if (a[j] > a[j + 1]) {

traodoi(a[j], a[j + 1]);

cotraodoi = true; } // kiem tra lan lap nay co doi cho khong

//neu khong co doi cho nao duoc thuc hien -> mang da sap xep. khong can

lap them

if (cotraodoi == false){

break;

}

}

}

( 1 **5 4** 2 8 ) -> ( 1 **4 5** 2 8 ), đổi chỗ do 5 > 4

( 1 4 **5 2** 8 ) -> ( 1 4 **2 5** 8 ), đổi chỗ do 5 > 2

( 1 4 **2 5 8** ) -> ( 1 4 2 **5 8** ), ở đây, hai phần tử đang xét đã đúng thứ tự

( 8 > 5 ) nên ta không cần đổi chỗ.

**Lần lặp thứ 2:**

(**14**258)->(**14**258)

( 1 **4 2** 5 8 ) -> ( 1 **2 4** 5 8 ), đổi chỗ do 4 > 2

(12**45**8)->(12**45**8)

(124**58**)->(124**58**)

**Code minh họa:**

void xuatmang(int a[], int size){

int i;

for (i = 0; i < size; i++)

printf("%d ", a[i]);

}

int main()

{

int a[] = { 5,1,4,2,8 };

int n = sizeof(a) / sizeof(a[0]);

SapXepNoiBot(a, n);

printf("mang da sap xep:");

xuatmang(a, n);

}

**3.2. Thuật toán sắp xếp chọn**

**Ý tưởng thuật toán sắp xếp chọn:**

Thuật toán sắp xếp chọn sẽ sắp xếp một mảng bằng cách đi tìm phần tử có giá trị nhỏ nhất(giả sử với sắp xếp mảng tăng dần) trong đoạn đoạn chưa được sắp xếp và đổi cho phần tử nhỏ nhất đó với phần tử ở đầu đoạn chưa được sắp xếp(không phải đầu mảng).

Thuật toán sẽ chia mảng làm 2 mảng con:

1. Một mảng con đã được sắp xếp
2. Một mảng con chưa được sắp xếp

Tại mỗi bước lặp của thuật toán, phần tử nhỏ nhất ở mảng con chưa được sắp xếp sẽ được di chuyển về đoạn đã sắp xếp.

**Ví dụ minh họa:**

Giả sử chúng ta cần sắp xếp dãy số [ 63 25 16 23 2 ] này ***tăng dần.***

Tìm phần tử nhỏ nhất trong mảng [0…4] và đổi chỗ nó với phần tử đầu tiên: **63** 25 16 23 **2** -> [**2]** 25 16 23 **63**

Tìm phần tử nhỏ nhất trong mảng [1…4] và đổi chỗ nó với phần tử đầu tiên của a[1…4]:2 **25 16** 23 63 -> 2 [**16] 25** 23 63

Tìm phần tử nhỏ nhất trong mảng [2…4] và đổi chỗ nó với phần tử đầu tiên của a[2…4]:

2 16 **25 23** 63-> 1 16 **[23] 25** 63

Tìm phần tử nhỏ nhất trong mảng [3…4] và đổi chỗ nó với phần tử đầu tiên của a[3…4]:

1 16 23 **25 63** -> 1 16 23 **[25] 63**

**Code minh họa 1:**

#include <stdio.h>

* Ham doi cho 2 so nguyen void TraoDoi(int &x, int &y){

int temp = x;

x= y;

x= temp;

}

// Hàm sap xep chen

void SapXepChon(int a[], int n){

int i, j, min;

//Di chuyen ranh gioi cua mang da sap xep va chua sap xep

for (i = 0; i < n - 1; i++){

//Tim phan tu nho nhat trong mang chua sap xep

min = i;

for (j = i + 1; j < n; j++)

if (a[j] < a[min])

min = j;

//Doi cho phan tu nho nhat voi phan tu dau tien TraoDoi(a[min], a[i]);

}

}

// ham xuat mang

void XuatMang(int a[], int size)

{

int i;

for (i = 0; i < size; i++)

printf("%d ", a[i]);

}

int main(){

int a[] = { 63, 25, 16, 23, 2 };

int n = sizeof(a) / sizeof(a[0]);

SapXepChen(a, n);

printf("Mang da sap xep: ");

XuatMang(a, n);

}

//Hàm Main:

void main()

{

int A[100];

NhapMang(A, N);

Sapxep(A, N);

printf("Mang sau khi sap xep la: ");

XuatMang(A, N);

getch();

}

**3.3. Thuật toán sắp xếp chèn**

**Ý tưởng thuật toán sắp xếp chèn:**

Thuật toán sắp xếp chèn thực hiện sắp xếp dãy số theo cách duyệt từng phần tử và chèn từng phần tử đó vào đúng vị trí trong mảng con(dãy số từ đầu đến phần tử phía trước nó) đã sắp xếp sao cho dãy số trong mảng sắp đã xếp đó vẫn đảm bảo tính chất của một dãy số tăng dần.

1. Khởi tạo mảng với dãy con đã sắp xếp có k = 1 phần tử(phần tử đầu tiên, phần tử có chỉ số 0)
2. Duyệt từng phần tử từ phần tử thứ 2, tại mỗi lần duyệt phần tử ở chỉ số i thì đặt phần tử đó vào một vị trí nào đó trong đoạn từ [0…i] sao cho dãy số từ [0…i] vẫn đảm bảo tính chất dãy số tăng dần. Sau mỗi lần duyệt, số phần tử đã được sắp xếp k trong mảng tăng thêm 1 phần tử.
3. Lặp cho tới khi duyệt hết tất cả các phần tử của mảng

**Ví dụ minh họa:**

1. Cho dẫy số từ trái sang phải:

5431113267

**54**31113267



**453**1113267



345**11**13267

34511**13**267

**34511132**67

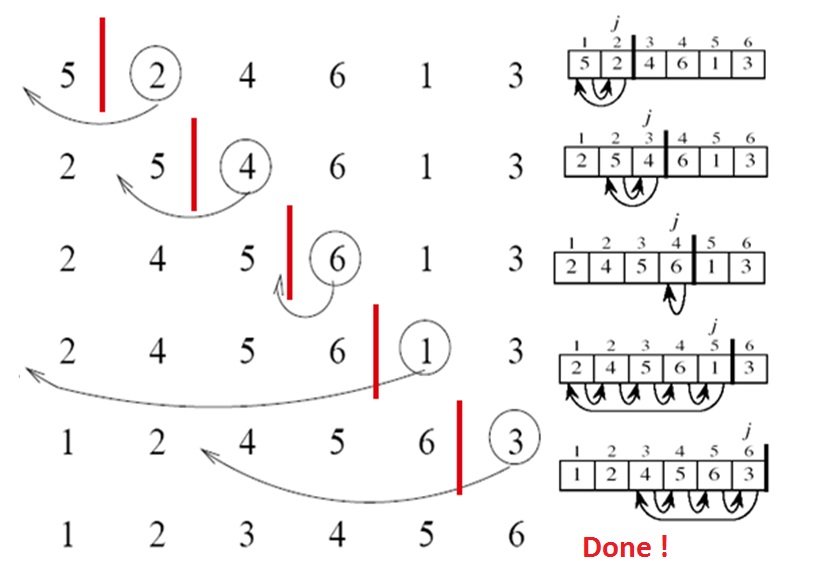
2345**11136**7



****23456**11137**

2345671113

1. Cho dẫy số từ trái sang phải:



Hình 3.1

**Code minh họa:**

#include <stdio.h>

#include <math.h>

* Ham thuat toan sap xep chen void SapXepChen(int a[], int n){

int i, key, j;

for (i = 1; i < n; i++){

key = a[i]; j = i - 1;

//di chuyen cac phan tu co gia tri lon hon gia tri key ve sau mot vi tri so voi vi tri ban dau cua no

while (j >= 0 && a[j] > key){

a[j + 1] = a[j]; j = j - 1;

}

a[j + 1] = key;

}

}

void XuatMang(int a[], int n){

int i;

for (i = 0; i < n; i++)

printf("%d ", a[i]);

printf("\n");

}

int main(){

int a[] = { 5,4,3,11,13,2,6,7 };

int n = sizeof(a) / sizeof(a[0]);

SapXepChen(a, n);

printf("Mang sap xep:")

XuatMang(a, n);

}

* 1. **Tìm kiếm tuyến tính**

**Ý tưởng thuật toán:**

Cho một mảng đã sắp xếp a[] có n phần tử, viết một hàm tìm kiếm trả về chỉ số của phần tử có giá trị x trong a[]. a

Giải thuật đơn giản nhất cho bài toán này là sử dụng linear search (tìm kiếm tuyến tính).

Tức là sẽ phải đi qua từng phần tử của mảng để đối chiếu với x cần tìm.

**Code minh họa:**

#include <stdio.h>

int TimKiem(int a[], int n, int x){

for (int i = 0; i < n; i++)

if (a[i] == x)

//Trả về chỉ số khi tìm thấy return i;

//Nếu không tìm thấy trả về -1.

return -1;

}

int main(){

int x = 50;

int a[] = { 2, 3, 4, 10, 40 };

int n = sizeof(a) / sizeof(a[0]);

int ketqua = TimKiem(a, n, x);

if (ketqua != -1){

printf("%d xuat hien tai chi so %d", x, ketqua);

}

else{

printf("%d khong co trong mang", x);

}

return 0;

}

* 1. **Tìm kiếm nhị phân**

**Ý tưởng thuật toán:**

* 1. Xét đoạn mảng a[trái…phai] cần tìm kiếm phần tử x. Ta so sánh x với phần tử ở vị trí giữa của mảng(giữa = (trái + phải)/2). Nếu:
  2. Nếu phần tử a[giữa] = x. Kết luận và thoát chương trình.
  3. Nếu a[giữa] < x. Chỉ thực hiện tìm kiếm trên đoạn a[giữa+1…phải].
  4. Nếu a[giữa] > x. Chỉ thực hiện tìm kiếm trên đoạn a[trái…giữa-1].

**Code minh họa:**

#include <stdio.h>

* Hàm tìm kiếm nhị phân sử dụng giải thuật đệ quy

int TimKiem(int a[], int t, int p, int x){

if (p >= t){

int giua = 1 + (p - t) / 2; // Tương đương (t+p)/2

//Nếu a[giua] = x, trả về chỉ số và kết thúc.

if (a[giua] == x)

return giua;

//Nếu a[giua] > x, thực hiện tìm kiếm nửa trái của mảng

if (a[giua] > x)

return TimKiem(a, t, giua - 1, x);

//Nếu a[giua] < x, thực hiện kiếm nửa phải của mảng

return TimKiem(a, giua + 1, p tìm, x);

}

//Nếu không tìm thấy return -1;

}

int main(){

int a[] = { 2, 3, 4, 10, 40 };

int n = sizeof(a) / sizeof(a[0]);

int x = 3;

int ketqua = TimKiem(a, 0, n - 1, x);

if (ketqua == -1)

printf("%d xuat hien tai chi so %d", x, ketqua);

else

printf("%d xuat hien tai chi so %d", x, ketqua);

return 0;

}

**III. Xây dựng chương trình code**

***Bài Code:***

#include<stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

typedef struct NhanVien {

char hoten[30];

char tinh [30];

int namsinh;

} NhanVien;

void nhap(NhanVien nhanvien[3]) {

for(int i=0; i<3; i++) {

fflush(stdin);

printf("\nNhap sinh vien thu %d: \n",i+1);

printf("\n Hoten: ");

gets(nhanvien[i].hoten);

printf("\n Tinh: ");

gets(nhanvien[i].tinh );

printf("\n Namsinh: ");

scanf("%d",&nhanvien[i].namsinh );

}

}

void sapXepHienThi(NhanVien nhanvien[3]) {

NhanVien temp;

for(int i=0; i<3; i++) {

for(int j=i+1; j<3; j++)

if(strcmp(nhanvien[i].tinh,nhanvien[j].tinh) < 0) {

temp=nhanvien[i];

nhanvien[i]=nhanvien[j];

nhanvien[j]=temp;

}

}

printf("STT||Hoaten ||Tinh ||Namsinh\n");

for(int i=0; i<3; i++) {

printf("00%d||%-20s||%-20s||%d\n",i+1,nhanvien[i].hoten,nhanvien[i].tinh,nhanvien[i] .namsinh);

}

}

void timNhanVien(NhanVien nhanvien[3]) {

char timnv[15];

fflush(stdin);

printf("\n Nhap tinh can tim :");

printf("\n Nhap tinh can tim :");

gets(timnv);

int t;

printf("STT||Hoten ||Tinh ||Namsinh\n");

for(int i=0; i<3; i++) {

t=0;

if(strcmp(timnv,nhanvien[i].tinh)==0) {

printf("00%d||%-20s||%-20s||%d\n",i+1,nhanvien[i].hoten,nhanvien[i].tinh,nhanvien[i]. namsinh);

t=1;

}

}

if(t==0) {

printf("\nKhong co nhan vien thuoc tinh nay.");

}

}

int main() {

FILE \*fp;

NhanVien nhanvien[3];

system("color a1");

while(1) {

int key , m=0;

system("cls");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\*\* \*\*\n");

printf("\t\t\t\*\* CHUONG TRINH QUAN LY NHAN VIEN \*\*\n");

printf("\t\t\t\*\* 1. Nhap du lieu \*\*\n");

printf("\t\t\t\*\* 2. Sap xep va hien thi thong tin chi \*\*\n");

printf("\t\t\t\*\* tiet cua tung nhan vien theo tinh tu Z->A \*\*\n");

printf("\t\t\t\*\* 3. Tim nhan vien theo tinh \*\*\n");

printf("\t\t\t\*\* 4. Ghi ra tap tin DSSV.txt \*\*\n");

printf("\t\t\t\*\* 5. Thoat \*\*\n");

printf("\t\t\t\*\* \*\*\n");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\*\* Nhap vao lua chon cua ban (1 den 5): \*\*\n");

scanf("%d",&key);

switch(key) {

case 1:

nhap(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 2:

sapXepHienThi(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 3:

timNhanVien(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 4:

fp=fopen("C:\\Users\\WELCOME\\Desktop\\DSNV.txt","a+");

if(fp==NULL) {

printf("File khong mo duoc");

}

for(int i=0; i<3; i++) {

fprintf(fp,"Nhan vien %d Hoten: %s Tinh:%s Namsinh:%d \n",i+1,nhanvien[i].hoten, nhanvien[i].tinh,nhanvien[i].namsinh);

}

fclose(fp);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 5:

exit(1);

return 0;

}

}

return 0;

}

**Giải thích code:**

## Ý Tưởng

Đầu tiên sẽ định nghĩa 1 kiểu dữ liệu mới là NhanVien sau đó khai báo các thành phần trong quản lí nhân sự.

//định nghĩa kiểu dữ liệu

typedef struct NhanVien {

char hoten[30];

char tinh [30];

int namsinh;

} NhanVien;

Sau đó chúng ta sẽ đến với các hàm sử dụng trong bài:

// các hàm sử dụng trong chương trình quản lí

* void nhap(NhanVien);

(*Hàm này cho phép ta nhập các thông số của nhân viên)*

* void sapXepHienThi(NhanVien);

(*Hàm này cho phép ta sắp xếp hiển thị thông tin nhân viên)*

* void timNhanVien(NhanVien);

(*Hàm này cho phép ta tìm kiếm nhân viên theo yêu cầu)*

## Chi tiết và giải thích code

- ***Đầu tiên***, tạo hàm **nhapNhanVien** để nhập thông tin nhân viên với số lượng được khai báo là 3 nhân viên.

void nhap(NhanVien nhanvien[3]) {

for(int i=0; i<3; i++) {

fflush(stdin);

printf("\nNhap sinh vien thu %d: \n",i+1);

printf("\n Hoten: ");

gets(nhanvien[i].hoten);

printf("\n Tinh: ");

gets(nhanvien[i].tinh );

printf("\n Namsinh: ");

scanf("%d",&nhanvien[i].namsinh );

}

}

- ***Thứ hai***, tạo hàm **sapXepHienThi** để sắp xếp và hiển thị thông tin nhân viên.

void sapXepHienThi(NhanVien nhanvien[3]) {

NhanVien temp;

for(int i=0; i<3; i++) {

for(int j=i+1; j<3; j++)

if(strcmp(nhanvien[i].tinh,nhanvien[j].tinh) < 0) {

temp=nhanvien[i];

nhanvien[i]=nhanvien[j];

nhanvien[j]=temp;

}

}

printf("STT||Hoaten ||Tinh ||Namsinh\n");

for(int i=0; i<3; i++) {

printf("00%d||%-20s||%-20s||%d\n",i+1,nhanvien[i].hoten,nhanvien[i]. tinh,nhanvien[i].namsinh);

}

}

* **Thứ ba**, tạo hàm timNhanVien để tìm kiếm nhân viên theo yêu cầu (tỉnh).

void timNhanVien(NhanVien nhanvien[3]) {

char timnv[15];

fflush(stdin);

printf("\n Nhap tinh can tim :");

gets(timnv);

int t;

printf("STT||Hoten ||Tinh ||Namsinh\n");

for(int i=0; i<3; i++) {

t=0;

if(strcmp(timnv,nhanvien[i].tinh)==0) {

printf("00%d||%-20s||%-20s||%d\n",i+1,nhanvien[i].hoten,nhanvien[i]. tinh,nhanvien[i].namsinh);

t=1;

}

}

if(t==0) {

printf("\nKhong co nhan vien thuoc tinh nay.");}

}

- *Cuối cùng*, ta sẽ tạo một hàm **Menu** có chức năng như liệt kê các hàm tạo ra sự lựa chọn các ý mà người dùng muốn lựa chọn tương ứng với các hàm mà ta đã xây dựng bên trên.

int main() {

FILE \*fp;

NhanVien nhanvien[3];

system("color a1");

while(1) {

int key , m=0;

system("cls");// hoan thanh xong chuc nag se tra ve printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\*\* \*\*\n");

printf("\t\t\t\*\* CHUONG TRINH QUAN LY NHAN VIEN \*\*\n");

printf("\t\t\t\*\* 1. Nhap du lieu \*\*\n");

printf("\t\t\t\*\* 2. Sap xep va hien thi thong tin chi \*\*\n");

printf("\t\t\t\*\* tiet cua tung nv theo tinh tu Z->A \*\*\n");

printf("\t\t\t\*\* 3. Tim nhan vien theo tinh \*\*\n");

printf("\t\t\t\*\* 4. Ghi ra tap tin DSSV.txt \*\*\n");

printf("\t\t\t\*\* 5. Thoat \*\*\n");

printf("\t\t\t\*\* \*\*\n");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\*\* Nhap vao lua chon cua ban (1 den 5): \*\*\n");

scanf("%d",&key);

switch(key) {

case 1:

nhap(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 2:

sapXepHienThi(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 3:

timNhanVien(nhanvien);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 4:

fp=fopen("C:\\Users\\WELCOME\\Desktop\\DSNV.txt","a+");

if(fp==NULL) {

printf("File khong mo duoc");

}

for(int i=0; i<3; i++) {

fprintf(fp,"Nhan vien %d Hoten: %s Tinh:%s Namsinh:%d \n",i+1,nhanvien[i]. hoten,nhanvien[i].tinh,nhanvien[i].namsinh);

}

fclose(fp);

printf("\nBam phim bat ky de tiep tuc!");

getch();

break;

case 5:

exit(1);

return 0;

}

}

return 0;

}

Chú ý: trong **case 4** tạo hàm để lưu các thông tin nhân viên vào tệp văn bản DSNV.txt, bao gồm mở, ghi và đóng file.

# 

# CHƯƠNG 4: HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

## 1.Menu chính

- Khi chạy ứng dụng, chương trình sẽ hiễn thị ra Menu chính bao gồm các chức năng sau:



- Sử dụng các số từ 1 đến 5 để chọn các chức năng có trong menu.

## Nhập dữ kiệu (Nhập số 1 => Enter)

- Ở đây chúng ta nhập thông tin của nhân viên bao gồm : họ tên, tỉnh và năm sinh.

## Sắp xếp và hiển thị thông tinh nhân viên ( Nhập số 2 =>Enter)

- Chương trình sẽ sắp xếp và hiển thị thông tin của các nhân viên đã khai báo từ trước theo yêu cầu (cụ thể là tỉnh từ Z->A).

## Tìm nhân viên(Nhập số 3 =>Enter)

- Chương trình sẽ tìm nhân viên theo yêu cầu(cụ thể là tỉnh).

## Ghi ra tập tin(Nhập số 4 =>Enter)

- Chương trình sẽ xuất ra màn hình tất cả thông tin theo yêu cầu trên vào tập tin DSNV.txt.

## Thoát chương trình(Nhập số 5 =>Enter)

- Chương trình sẽ tự động dừng lại.

**CHƯƠNG 5: TỔNG KẾT**

**1.Kết quả đạt được**

Ưu điểm và khuyết điểm:

* **Ưu điểm:**
* Từ mô hình quản lí nhân viên. Thông qua đó chúng ta có thể dùng mô hình này để xử lí cho những mô hình có cách tổ chức tương tự như quản lí học sinh trường học, quản lí các học viên trung tâm ,...
* Mô hình đã đưa ra một số chức năng phù hợp với yêu cầu cần thiết để tiến hành quản lí sinh viên hiệu quả và chặt chẽ hơn.
* Cấu trúc dữ liệu tương đối đầy đủ và gần gũi với thực tế giúp mọi người nắm bắt được một cách dễ dàng
* **Khuyết điểm:**
* Hệ thống dữ liệu chưa thật sự đầy đủ
* Giao diện thiết kế chưa thẩm mĩ

**2.Hướng phát triển và hoàn thiện**

* Chương trình chỉ mới cho phép số ít người dùng tiếp cận với hệ thống, nên hướng phát triển là cần phải tối ưu việc phân quyền sử dụng như đổi mật khẩu tự động... cho người dùng có thể tiếp cận với các chức năng đáp ứng yêu cầu sử dụng.
* Bổ sung thêm nhiều chức năng nữa cho phù hợp với công việc quản lí nhân viên : như đăng kí ca làm, lịch làm việc,...
* Tối ưu chương trình và xây dựng hệ thống có thể thực hiện với việc sử dụng Online cho việc tiếp cận hệ thống của nhân viên.

**3.Tài liệu tham khảo**

[1]<https://hiepsiit.com/index.php/khoa-hoc/c/laptrinhc/50/0>

[2]<https://nguyenvanhieu.vn/thuat-toan-sap-xep-noi-bot/?fbclid=IwAR3IJtPd->[QvX5Bys7zrbEyMdpUvkM0EUj6plf85cYk7IkDd6iSQGwo32\_ow](https://nguyenvanhieu.vn/thuat-toan-sap-xep-noi-bot/?fbclid=IwAR3IJtPd-QvX5Bys7zrbEyMdpUvkM0EUj6plf85cYk7IkDd6iSQGwo32_ow)

[3][https://nguyenvanhieu.vn/thuat-toan-sap-xep-chon/?fbclid=IwAR1IrC1q1Lnu5msd4wNd57E9ps7Yd4tpq87H7gbIVGp8x3wL6uMip](https://nguyenvanhieu.vn/thuat-toan-sap-xep-chon/?fbclid=IwAR1IrC1q1Lnu5msd4wNd57E9ps7Yd4tpq87H7gbIVGp8x3wL6uMipWWmpf0) [WWmpf0](https://nguyenvanhieu.vn/thuat-toan-sap-xep-chon/?fbclid=IwAR1IrC1q1Lnu5msd4wNd57E9ps7Yd4tpq87H7gbIVGp8x3wL6uMipWWmpf0)

[4][https://nguyenvanhieu.vn/thuat-toan-sap-xep-chen/?fbclid=IwAR0ZeTgGDw7Jzsmh70fAI3Ful9ah6Qe31opsHwiC3fNZlE\_KhmnT\_o](https://nguyenvanhieu.vn/thuat-toan-sap-xep-chen/?fbclid=IwAR0ZeTgGDw7Jzsmh70fAI3Ful9ah6Qe31opsHwiC3fNZlE_KhmnT_ozEtNE) [zEtNE](https://nguyenvanhieu.vn/thuat-toan-sap-xep-chen/?fbclid=IwAR0ZeTgGDw7Jzsmh70fAI3Ful9ah6Qe31opsHwiC3fNZlE_KhmnT_ozEtNE)

[5]<https://nguyenvanhieu.vn/thuat-toan-tim-kiem-nhi-phan/?fbclid=IwAR3_yWK2LNlkYS1_2rYvNvx_460WR-Nw8zjAOXJZfbhZ7_qHMQZRwRm8vvk>

[6]<https://nguyenvanhieu.vn/doc-ghi-file-trong-c/?fbclid=IwAR1Ly-mZ603lfm_KTnhZ9ovwhXz538mZk1wX_qCB_pHeoHCw5pTyGqy01EA>

[7]<https://sinhvientot.net/thuat-toan-sap-xep-noi-bot-bubblesort/>

[8]<https://sinhvientot.net/thuat-toan-sap-xep-chon-truc-tiep-selection-sort/>

[9]<https://vietjack.com/lap_trinh_c/file_io_trong_c.jsp>

[10]<https://vietjack.com/bai-tap-c/tao-danh-sach-lien-ket-don-trong-c.jsp>

[11]<https://nguyenvanhieu.vn/danh-sach-lien-ket-don/>

[12]<https://nguyenvanhieu.vn/chuong-trinh-quan-ly-sinh-vien-su-dung-struct/>

**KẾT LUẬN**

Trong giới hạn thời gian cho phép, chúng em đã cố gắng vận dụng lí thuyết đã học để giải quyết yêu cầu của đề tài. Tuy nhiên vì trình độ có hạn và sự nhìn nhận chưa đủ, nên báo cáo khó tránh khỏi những hạn chế và thiếu sót nhất định. Với tinh thần tìm tòi học hỏi, chúng em rất mong nhận được sự góp ý chỉ bảo của các thầy cô để chương trình ngày một hoàn thiện hơn.

Chúng em xin chân thành cảm ơn sự giúp đỡ của các thầy cô giáo trong khoa Công nghệ thông tin trường Đại học Giao thông Vận tải phân hiệu tại TP.Hồ Chí Minh đã truyền đạt kiến thức trong thời gian học tại trường , đặc biệt là cô Trần Thị Dung đã tận tình hướng dẫn, giúp đỡ chúng em trong suốt qua trình làm báo cáo để chúng em có thể hoàn thiện đề tài quản lí nhân viên này.

**Chúng** **em xin chân thành cảm ơn!**

**Võ Quang Vinh**

**Bùi Minh Thiện**

**Huỳnh Xuân Lãm**