

**-**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

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



**BÁO CÁO BÀI TẬP LỚN**

**Môn: Lập trình nâng cao**

**Đề tài: Chương trình nhập xuất dữ liệu thành phố**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Giảng viên hướng dẫn:** | Trần Thị Dung |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Sinh viên:** | **MSSV:** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Vũ Hoàng Phi | 6051071091 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Phan Quang Đông | 6051071031 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Nguyễn Lâm Trúc | 6051071131 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Mã học phần:** | CPM215.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**LỜI CẢM ƠN**

Tập thể nhóm F3 chúng em xin gửi lời cảm ơn chân thành đối với cô Trần Thị Dung, cảm ơn cô trong thời gian qua đã tận tình, tận tâm giúp đỡ, hướng dẫn và luôn giao rất nhiều bài tập cho chúng em làm để nhóm có thể hoàn thành báo cáo bài tập lớn môn “ Lập trình nâng cao”.

Sau một khoảng thời gian theo học tại trường Đại học Giao thông vận tải phân hiệu tại TPHCM, chúng em đã được tiếp xúc và rèn luyện với lập trình. Những khoảng thời gian ấy chúng em không ngừng học hỏi để có thể phát triển bản thân mình. Trong quá trình làm bài tập lớn và báo cáo, vì đây là lần đầu tiên nhóm em làm báo cáo nên đôi khi kiến thức của nhóm còn những hạn chế nhất định, do vậy sẽ không tránh khỏi thiếu sót, đôi khi sẽ có những ấp úng trong lúc vấn đáp nhưng nhóm F3 chúng em rất mong nhận được ý kiến đóng góp của cô để chúng em có thể dần hoàn thiện hơn.

Lời cuối cùng, nhóm F3 chúng em một lần nữa xin gửi lời cảm ơn chân thành đến cô và nhóm xin chúc cô Trần Thị Dung sẽ luôn mạnh khỏe và ngày càng xinh đẹp hơn nữa.

Trân trọng!

Tp. Hồ Chí Minh, ngày 20 tháng 6 năm 2020 Sinh viên thực hiện

Vũ Hoàng Phi

Phan Quang Đông

Nguyễn Lâm Trúc

**Mục lục**

**Chương I. TỔNG QUAN ĐỀ TÀI**

* 1. **Lí do chọn đề tài….…………………………………………………………………….. 1**
  2. **Đối tượng nghiên cứu.……………………………………………………………….… 2**
  3. **Qúa trình nghiên cứu……………….………………………………………………….. 2**

**Chương II. LÝ THUYẾT ỨNG DỤNG**

**2.1. Làm việc với tệp………………………………………………………………………... 3**

**2.2. Danh sách liên kết đơn……………………………………………………………..….. 4**

**2.2.1. Các kiểu danh sách liên kết………………………………………………………. 5**

**2.2.2. So sánh mảng với danh sách liên kết đơn……………………………………… 5**

**2.2.3. Cài đặt danh sách liên kết đơn…………………………………………………… 6**

**2.2.4. Một số thao tác với danh sách liên kết đơn……………………………………… 7**

**2.3. Các thuật toán sắp xếp……………………………...………………………………... 11**

**2.3.1. Sắp xếp nổi bọt………………………………………………………………….... 11**

**2.3.2. Sắp xếp chọn trực tiếp…………………………………………………………… 13**

**2.3.3. Sắp xếp đổi trực tiếp………………………………………………………….….. 14**

**2.4. Các thuật toán tìm kiếm……………………………………………………………… 15**

**2.4.1. Tìm kiếm tuyến tính……………………………………………………………... 15**

**2.4.2. Tìm kiếm nhị phân………………………………………………………………. 16**

**Chương III: GIỚI THIỆU CHƯƠNG TRÌNH…………………………………………….... 18**

**3.1. Giới thiệu chương trình……………………………………………………………… 18**

**3.2. Giải thích thuật toán…………………………………………………………………. 18**

**3.3. Hướng dẫn sử dụng………………………………………………………………….. 23**

**Chương IV: TỔNG KẾT…………………………….………………………………………. 27**

**Chương V: TƯ LIỆU THAM KHẢO……………………………………………………….. 28**

**CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

* 1. **Lí do chọn đề tài:**

- Ngày nay, trên phạm vi toàn cầu với sự bùng nổ của khoa học công nghệ, sự phát triển không ngừng của kinh tế, thương mại và mức sống của người dân ngày càng được nâng cao đã gắn kết hoạt động du lịch, giải trí của con người với công việc làm ăn. Hàng năm, những sự kiện chính trị, kinh tế, văn hoá, thể thao mang tầm quốc gia, khu vực và quốc tế diễn ra với mật độ dày đặc đã trở thành một cơ hội tốt cho hoạt động du lịch nói chung phát triển và cũng trở thành mảnh đất màu mỡ để du lịch phát triển.

**-**  Khoa học công nghệ là thứ không thể thiếu đối với mỗi quốc gia, doanh nghiệp, trường học hay mỗi cá nhân, đặc biệt là công nghệ thông tin. Với sự phát triển nhanh một cách không ngừng nghĩ như vậy của công nghệ thông tin đã giúp giải quyết các công việc học tập, nguyên cứu, quản lý thông tin,… một cách dễ dàng và tiện lợi. Thấy được tiềm năng đó các quốc gia, doanh nghiệp, trường học, các cá nhân, … đã ứng dụng nó vào thực tiễn cuộc sống để giải quyết công việc, học tập, giải trí.

- Trong những năm gần đây nhu cầu về các phần mềm, ứng dụng học tập, giải trí càng nhiều hơn do nhu cầu các thiết bị công nghệ gọn nhẹ ngày càng cao và thuận tiện cho người sử dụng. Các thiết bị thông minh đã và đang được sử dụng rộng rãi trên toàn thế giới, từ trẻ em cho đến cụ già ai cũng có ít nhất là một chiếc điện thoại thông minh, máy tính và có thể thấy rằng các thiết bị công nghệ đang dần thay thế cho sách . Nắm bắt được tình hình đó hàng loạt nhà sản xuất đã đánh vào thị trường ứng dụng để phát triển. Người sử dụng có thể học tập và giải trí ở bất kì đâu mà họ muốn chỉ thông qua một chiếc điện thoại di động có kết nối với Internet. Đã có rất nhiều ứng dụng đã ra đời và đem lại nhiều lợi ích cho người sử dụng, bạn không cần phải đến thư viện để đọc sách hoặc tốn nhiều thời gian cho việc tìm kiếm thông tin hoặc là bạn không cần đến trường, trung tâm để học chỉ cần cài được ứng dụng bạn có thể gói gọn mọi thứ trong chiếc điện thoại của bạn.

- Trong du lịch, đối với cá nhân hay một tập thể nào đó, việc tìm hiểu trước nơi mà mình sắp đến là việc hết sức cần thiết. Việc này sẽ giúp chúng ta định hình rõ nơi ta đến, những địa điểm sắp tham quan lịch trình sẽ được lên rõ ràng hơn là khi đến đó chúng ta mới lên kế hoạch. Như vậy lịch trình có thể sẽ không suôn sẻ và tốn thời gian hơn việc ta đã lên kế hoạch từ trước. Chính vì thế mà các ứng dụng hay các trang web dùng để tìm kiếm thông tin ra đời. Chương trình nhập xuất dữ liệu thành phố là một phần tất yếu của các ứng dụng và các trang web tìm kiếm và là tiền đề để xây dựng mạnh mẽ hơn.

* 1. **Đối tượng nghiên cứu:**

**-** Chương trình nhập xuất dữ liệu thành phố là tiền đề để phát triển phần mềm tìm kiếm thông tin.

* 1. **Mục tiêu nghiên cứu:**

- Xây dựng được một chương trình có thể tìm kiếm thông tin đơn giản nhất.

- Chương trình phải có đầy đủ các chức năng để người dùng dễ sử dụng, giao diện phải liên quan đến vấn đề chính đó là “thành phố” để tạo cảm giác thích hợp cho người dùng.

**CHƯƠNG II: LÍ THUYẾT ỨNG DỤNG**

**2.1 Làm việc với tệp:**

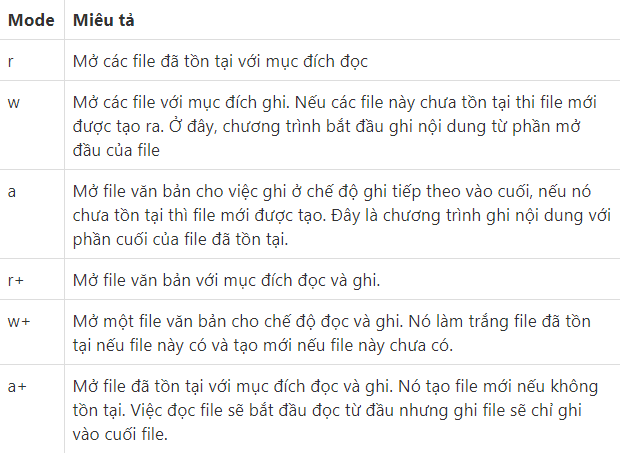
Trong ngôn ngữ lập trình C, có một số thao tác chính khi làm việc với file, bao gồm cả file văn bản và file nhị phân:

1. Tạo mới một file
2. Mở một file đã có
3. Đóng file đang mở
4. Đọc thông tin từ file/ Ghi thông tin ra file

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

VD: FILE \*p;

**Mở file:** fptr = fopen("fileopen","mode");  
Trong đó "mode" là một tham số chúng ta cần chỉ định, mà các mode chúng ta hay sử dụng trong C như “r”, “r+”, “w”, “w+”.

Hình 2.1

**Đóng file:** Sau khi thao tác với file xong, chúng ta cần phải đóng file lại

fclose(biến\_con\_trỏ\_tệp);

VD: fclose(fp);

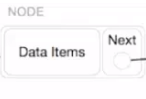
**Đọc/ghi file văn bản:** sử dụng fprintf() và fscanf().

**Đọc/ghi file nhị phân:** sử dụng fwrite() và fread().

**2.2. Danh sách liên kết đơn:**

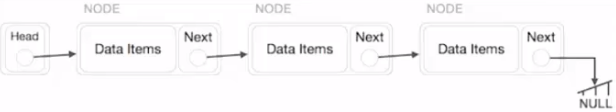
Danh sách liên kết đơn là một tập hợp các Node được phân bố động, được sắp xếp theo cách sao cho mỗi Node chứa “một giá trị”(Data) và “một con trỏ”(Next). Con trỏ sẽ trỏ đến phần tử kế tiếp của danh sách liên kết đó. Nếu con trỏ mà trỏ tới NULL, nghĩa là đó là phần tử cuối cùng của linked list.

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



Hình 2.2

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



Hình 2.3

**2.2.1. Các kiểu danh sách liên kết:**

Danh sách liên kết đơn(Single linked list): Chỉ có sự kết nối từ phần tử phía trước tới phần tử phía sau.

Danh sách liên kết đôi(Double linked list): Có sự kết nối 2 chiều giữa phần tử phía trước với phần tử phía sau

Danh sách liên kết vòng(Circular Linked List): Có thêm sự kết nối giữa 2 phần tử đầu tiên và phần tử cuối cùng để tạo thành vòng khép kín.

**2.2.2. So sánh mảng vs danh sách liên kết đơn:**

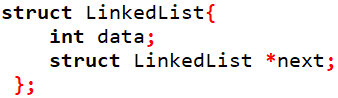
- Về bản chất, danh sách liên kết có chức năng như một mảng, có thể thêm và xóa các phần tử ở bất kỳ vị trí nào khi cần thiết. Một số sự khác nhau giữa danh sách liên kết và mảng:



Hình 2.4

**2.2.3. Cài đặt danh sách liên kết đơn:**

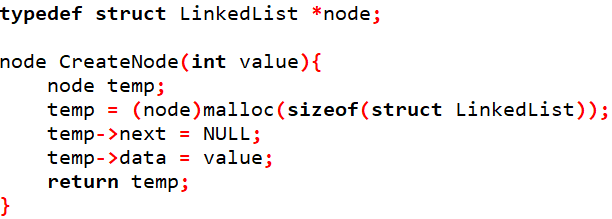
**B1: Khai báo LinkedList**



Hình 2.5

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 thằng 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.

**B2: Tạo mới 1 Node**



Hình 2.6

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, bạn chỉ cần lưu giữ Node đầu tiên (HEAD). Có head rồi bạn có thể đi tới bất cứ Node nào.

**2.2.4. Một số thao tác với danh sách liên kết đơn:**

Trong danh sách liên kết đơn, các Node sẽ không được lưu liên tiếp nhau trên bộ nhớ, Node trước sẽ mang thông tin địa chỉ của Node sau, như vậy nếu bạn xử lý lỗi một Node sẽ dẫn đến tính huống xấu nhất, ta sẽ mất toàn bộ thông tin của các Node phía sau.

Chúng ta sẽ bắt đầu làm việc với các thao tác cơ bản trên một danh sách liên kết đơn. Giả sử em có định nghĩa sau:

**struct SNode{**

**int Data;**

**SNode\* pNext;**

**};**

**struct SList{**

**SNode\* pHead;**

**SNode\* pTail;**

**SList(){}**

**SList(SNode\* Head, SNode\* Tail){**

**this->pHead = Head;**

**this->pTail = Tail;**

**}**

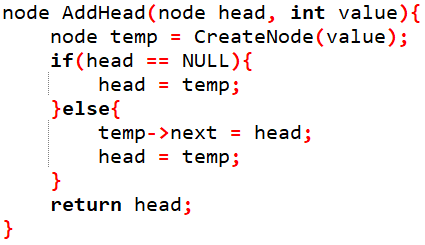
**};**

Nếu biết được địa chỉ đầu tiên trong danh sách liên kết ta có thể dựa vào thông tin pNext để truy xuất đến các phần tử còn lại, do đó ta sẽ dùng một con trỏ pHead để lưu lại địa chỉ Node đầu tiên của danh sách. Trong một số trường hợp ta cũng cần thao tác trên phần tử cuối cùng của danh sách, nên tôi dùng thêm một con trỏ pTail để lưu trữ địa chỉ của Node cuối cùng trong danh sách.

**- Thêm Node vào danh sách liên kết:**

**Thêm vào đầu**

Như đã trình bày ở trên, khi thao tác với mỗi Node trên danh sách liên kết ta cần thực hiện cẩn thận, đúng thứ tự để tránh mất thông tin của các Node phía sau. Dưới đây là thứ tự các bước chèn một phần tử vào đầu mảng.



Hình 2.7

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

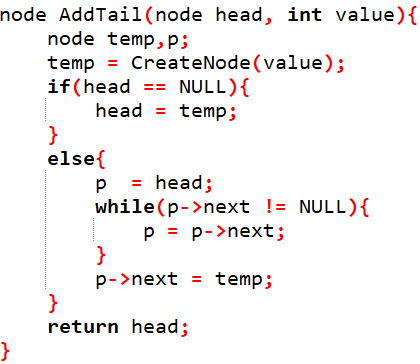
Ngược lại, ta phải thay thế head cũ bằng head mới. Việc này phải làm theo thứ tự như sau:

Cho next của temp trỏ tới head hiện hành.

Đặt temp làm head mới.

**Thêm vào cuối**

Tương tự với thao tác chèn vào đầu danh sách, chèn vào cuối danh sách chúng ta chỉ cần cập nhập lại con trỏ.



Hình 2.8

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 thằng cuối tới Node mới(temp).

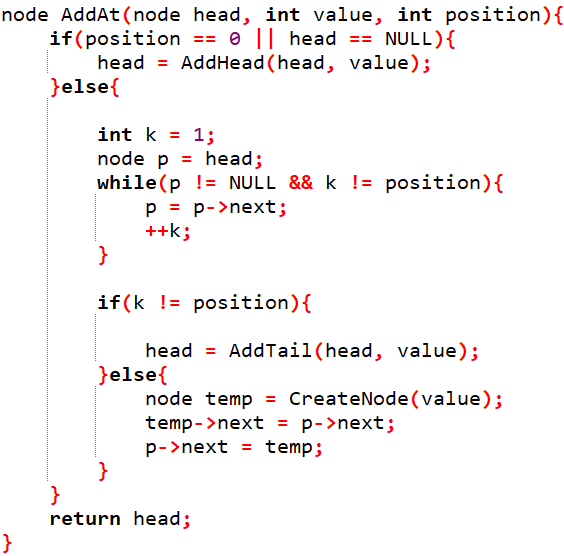
**Thêm vào vị trí bất kỳ:**

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 nhé các bạn



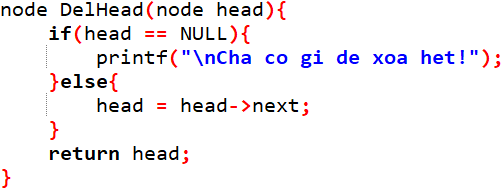
Hình 2.9

Lưu ý: Bạn phải làm theo thứ tự trên, nếu bạn cho p->next = temp trước. Khi đó, bạn 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)

**- Xoá node khỏi danh sách liên kết**

**Xóa đầu**

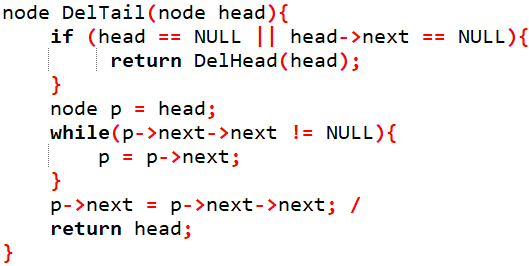
Cho giá trị kế tiếp của head làm head. Giá trị kế tiếp của head chính là head->next.



Hình 2.10

**Xoá cuối**

Duyệt đến giá trị cuối – 1, cho next của cuối – 1 đó bằng NULL.



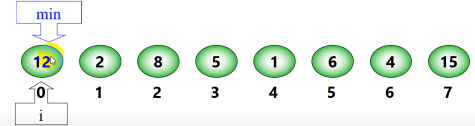
2.11

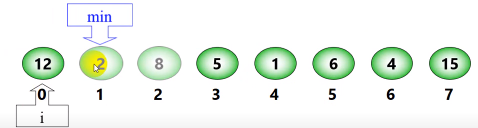
Giá trị Node cuối – 1 là giá trị có p->next->next = NULL. Cho next của nó bằng NULL là xong.

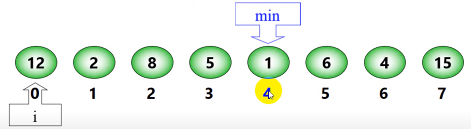
**2.3. Các thuật toán sắp xếp:**

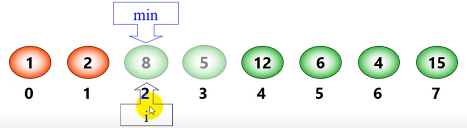
**2.3.1. Sắp xếp chọn trực tiếp:** Tìm ra phần từ nhỏ nhất trong tập T ban đầu, và hoán đổi vị trí của nó cho phần tử đầu tiên. Như vậy phần tử đầu tiên là phần tử có giá trị nhỏ nhất. Ta không quan tâm đến nó nữa bắt đầu 1 tập con tính từ phần tử thứ hai, lại làm tương tự như trên, tìm phần tử nhỏ nhất trong tập con và hoán vị nó về vị trí ban đâu của tập.

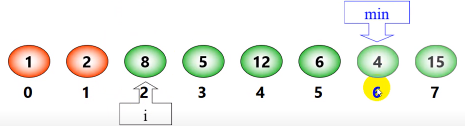
=> Cứ như vậy cho đến khi chuỗi chỉ còn 1 phần tử thì dừng lại.

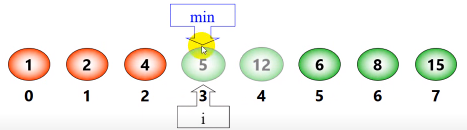


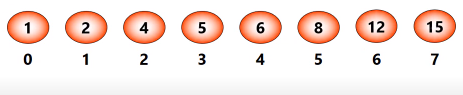
 Gán tạm thời cho 2 là GTNN

 Tìm thấy số 1 là GTNN liền trực tiếp đổi

 Thực hiện vòng lặp

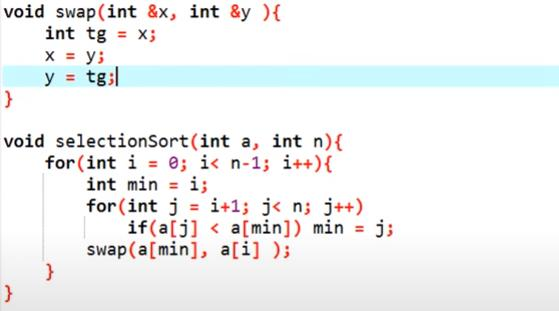
 Thực hiện vòng lặp

 Thực hiện vòng lặp

 Cho đến khi sắp xếp xong

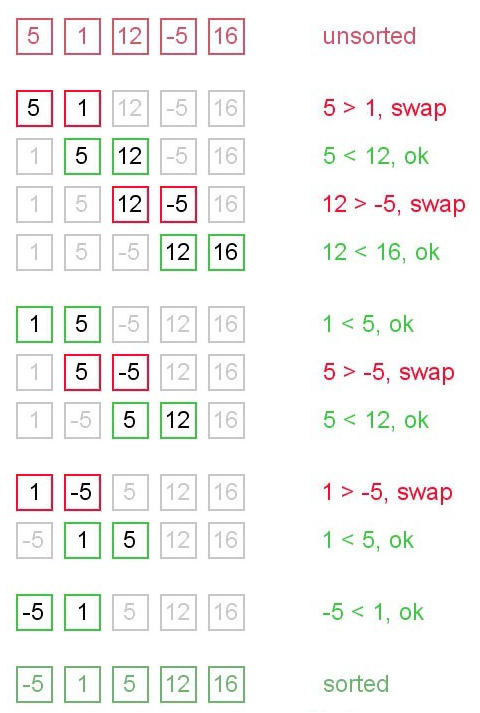
Hình 2.12

**Code VD:**



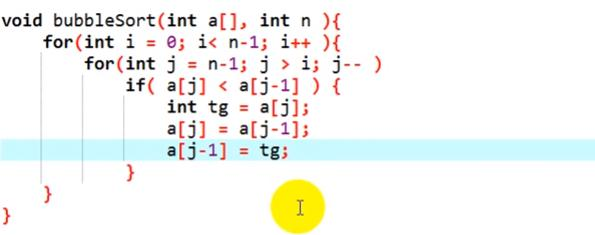
Hình 2.13

**2.3.2. Sắp xếp nổi bọt:** Phương pháp này sẽ duyệt danh sách nhiều lần, trong mỗi lần duyệt sẽ lần lượt so sánh cặp nút thứ i và thứ i+1 và đổi chỗ hai nút này nếu chúng không đúng thứ tự.



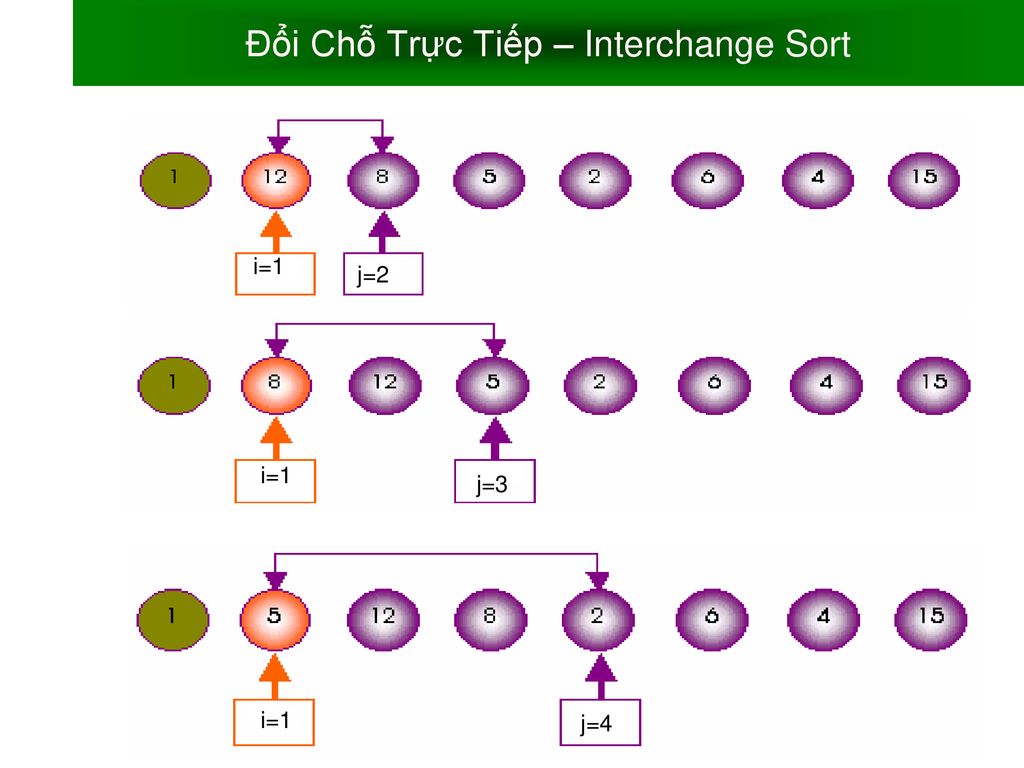
Hình 2.14

**Code VD:**



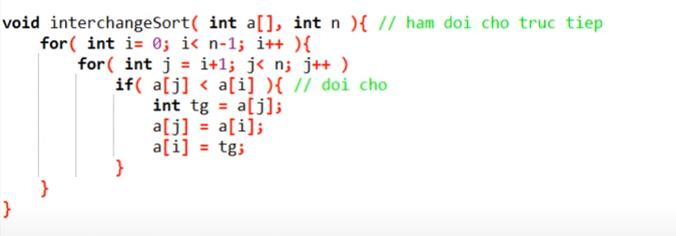
Hình 2.15

**2.3.3. Sắp xếp đổi trực tiếp:** Ý tưởng chính của giải thuật là xuất phát từ đầu dãy, tìm tất cả nghịch thế chứa phần tử này, triệt tiêu chúng bằng cách đổi chổ phần tử này với phần tử tương ứng trong cặp nghịch thế. Lặp lại xử lý trên với các phần tử kế tiếp theo trong dãy.



Hình 2.16

**Code VD:**



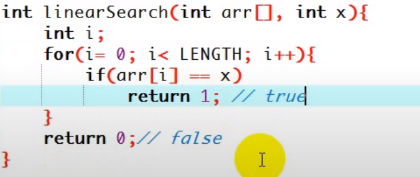
Hình 2.17

**2.4. Các thuật toán tìm kiếm:**

**2.4.1. Tìm kiếm tuyến tính (Linear Search)**Mỗi phần tử đều được kiểm tra và nếu tìm thấy bất kỳ kết nối nào thì phần tử cụ thể đó được trả về; nếu không tìm thấy thì quá trình tìm kiếm tiếp tục diễn ra cho tới khi tìm kiếm hết dữ liệu.

 Hình 2.18

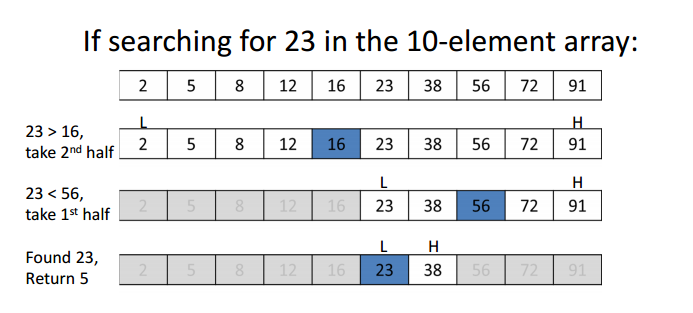
**Code VD:**



Hình 2.19

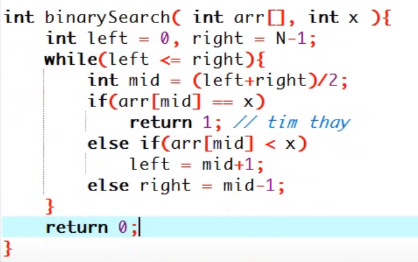
**2.4.2. Tìm kiếm nhị phân (Binary Search)**

Tìm kiếm nhị phân so sánh giá trị đích với phần tử ở giữa của mảng. Nếu chúng không bằng nhau, một nửa trong đó không chứa mục tiêu tìm kiếm bị loại bỏ và việc tìm kiếm tiếp tục ở nửa còn lại, một lần nữa lấy phần tử ở giữa được chọn để so sánh với giá trị đích và lặp lại điều này cho đến khi tìm thấy giá trị đích. Nếu tìm kiếm kết thúc với nửa còn lại trống, mục tiêu không nằm trong mảng.



Hình 2.20

**Code VD:**



Hình 2.21

**CHƯƠNG III: GIỚI THIỆU CHƯƠNG TRÌNH**

**3.1. Giới thiệu chương trình:**

Chương trình giúp cho người dùng dễ dàng nhập và xuất dữ liệu. Chương trình gồm nhiều chức năng như là: nhập dữ liệu thành phố, hiển thị danh sách thành phố từ A đến Z,hiển thị danh sách thành phố từ Z đến A, tìm thành phố theo tên quốc gia, ghi vào tập tin nhị phân.

**3.2. Giải thích thuật toán**

Đoạn code này được sử dụng để định nghĩa kiểu dữ liệu **city\_st** gồm nhiều thuộc tính như: **city\_nm** (Tên thành phố), **country** (Tên quốc gia của thành phố đó), **year** (Năm thành lập).

**typedef struct city\_st{**

**char city\_nm[30];**

**char country[30];**

**int year;**

**}CT;**

Hai hàm **nhap** và **nhapN** có chức năng dùng để nhập dữ liệu **n** thành phố (**n** là số nguyên được nhập từ bàn phím). Trong đó hàm **nhap**  được dùng để định nghĩa cấu trúc cần nhập cho 1 thành phố, còn hàm **nhapN** được sử dụng để lặp lại hàm **nhap** cho tới **n** thành phố.

**void nhap(CT &ct){**

**fflush(stdin);**

**printf("\nNhap ten thanh pho: ");**

**gets(ct.city\_nm);**

**printf("\nNhap ten quoc gia: ");**

**gets(ct.country);**

**printf("\nNhap nam: ");**

**scanf("%d", &ct.year);**

**}**

**void nhapN(CT a[], int n){**

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

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

**printf("\nNhap TP thu %d:", i+1);**

**nhap(a[i]);**

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

**}**

**}**

Hai hàm **xuat** và **xuatN** có chức năng dùng để hiển thị dữ liệu **n** thành phố đã được nhập ra ngoài màn hình console (**n** là số nguyên được nhập từ bàn phím). Trong đó hàm **xuat**  được dùng để định nghĩa cấu trúc cần hiển thi ra ngoài màn hình cho 1 thành phố, còn hàm **xuatN** được sử dụng để lặp lại hàm **xuat** cho tới **n** thành phố.

**void xuat(CT ct){**

**printf("\nTen: %s", ct.city\_nm);**

**printf("\nQuoc Gia: %s", ct.country);**

**printf("\nNam: %d", ct.year);**

**}**

**void xuatN(CT a[], int n){**

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

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

**printf("\nThong tin Thanh Pho thu %d:\n", i+1);**

**xuat(a[i]);**

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

**}**

**}**

Hàm **sapxepZtoA** để sắp xếp lại dữ liệu thành phố đã nhập theo thứ tự từ Z đến A.

**void sapxepZtoA(CT a[], int n){**

**CT t;**

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

**for(int j = i+1; j < n;++j){**

**if(strcmp(a[i].city\_nm,a[j].city\_nm)<0){**

**t = a[i];**

**a[i] = a[j];**

**a[j] = t;**

**}**

**}**

**}**

**}**

Hàm **sapxepAtoZ** để sắp xếp lại dữ liệu thành phố đã nhập theo thứ tự từ A đến Z.

**void sapxepAtoZ(CT a[], int n){**

**CT t;**

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

**for(int j = i+1; j < n;++j){**

**if(strcmp(a[i].city\_nm,a[j].city\_nm)>0){**

**t = a[i];**

**a[i] = a[j];**

**a[j] = t;**

**}**

**}**

**}**

**}**

Hàm **timkiem** có chức năng tìm kiếm và hiển thị thông tin tất cả các thành phố thuộc quốc gia vừa nhập từ bàn phím.

**void timkiemN(CT a[], int n){**

**int dem=0;**

**char ten[30];**

**fflush(stdin);**

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

**printf("\nNhap ten quoc gia cua thanh pho ban muon tim: \n");**

**gets(ten);**

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

**if(strcmp(a[i].country, ten)==0){**

**dem = ++dem;**

**}**

**}**

**if(dem==0){**

**printf("\nKhong co quoc gia nay trong du lieu!!!");**

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

**}**

**else {**

**printf("Co %d ket qua duoc tim thay", dem);**

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

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

**if(strcmp(a[i].country, ten)==0){**

**printf("TP thu %d:\n", i);**

**timkiem(a[i]);**

**}**

**}**

**}**

**}**

Hàm **xuatFile** dùng để ghi dữ liệu ra tập tin nhị phân city.dat

**void xuatFile(CT a[], int n, char fileName[]){**

**FILE \* fp;**

**fp = fopen (fileName,"wb");**

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

**fwrite(&a[i], sizeof(struct city\_st), 2, fp);**

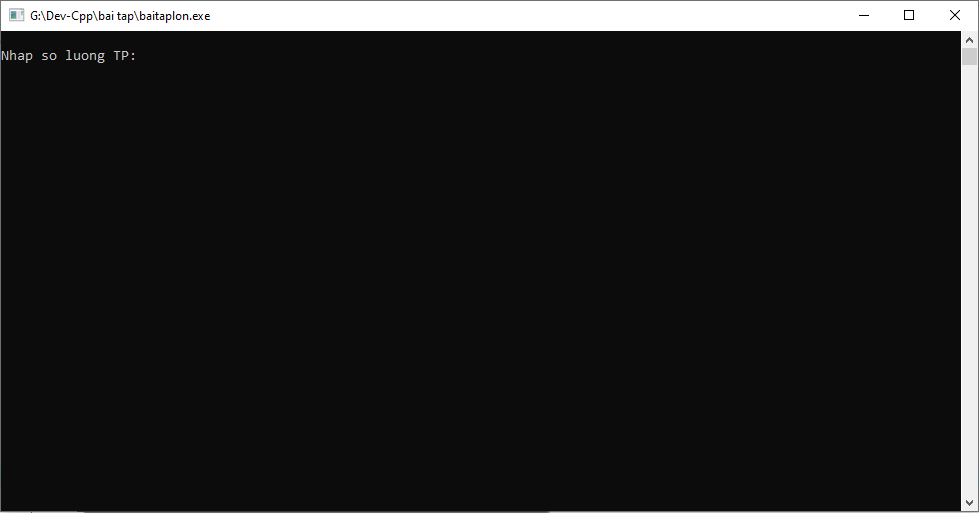
**}**

**fclose (fp);**

**}**

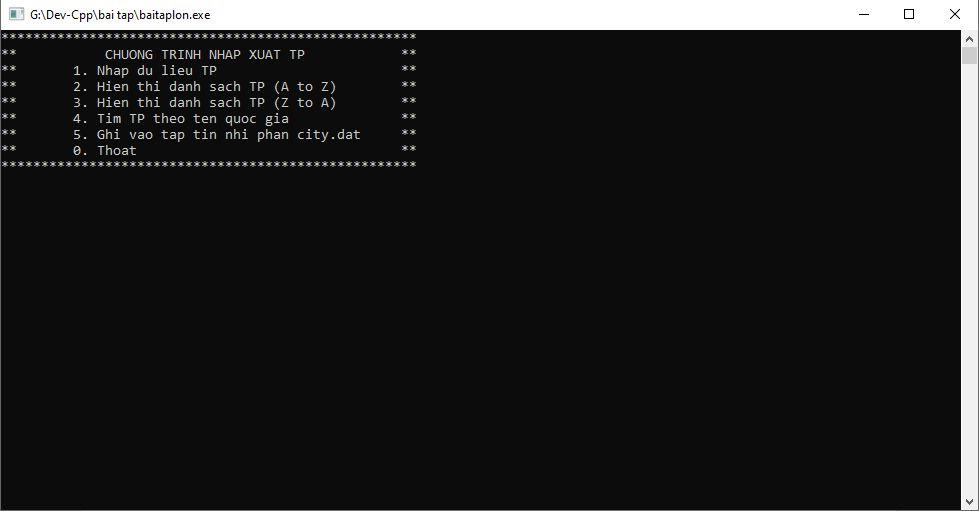
**3.3. Hướng dẫn sử dụng chương trình:**

Đầu tiên khi ta mở chương trình lên sẽ hiện một màn hình console để chúng ta có thể nhập số lượng thành phố mà chúng ta muốn nhập vào.



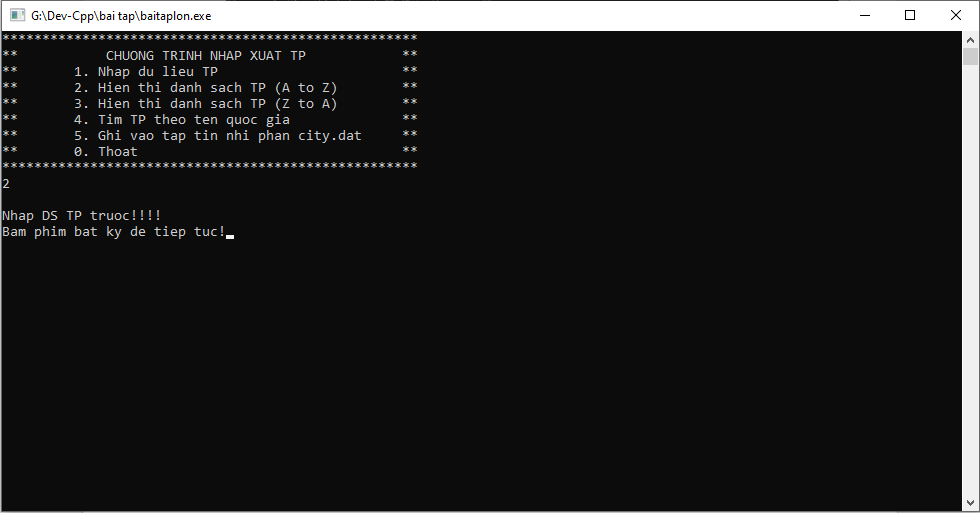
Hình 3.1

Sau khi nhập xong, chương trình sẽ hiện lên menu chức năng để ta có thể thực hiện thao tác mình muốn.



Hình 3.2

**LƯU Ý:** Trước khi sử dụng thì đầu tiên bạn phải nhập dữ liệu cho các thành phố bằng cách chọn số 1, nếu không các chức năng khác sẽ không hoạt động vì chưa có dữ liệu.



Hình 3.3

Sau khi đã nhập dữ liệu cho các thành phố thì bạn đã có thể sử dụng tất cả các chức năng của menu:

Chọn số 2: Chương trình sẽ hiển thị danh sách các thành phố đã nhập ra màn hình console theo thứ tự từ A tới Z.



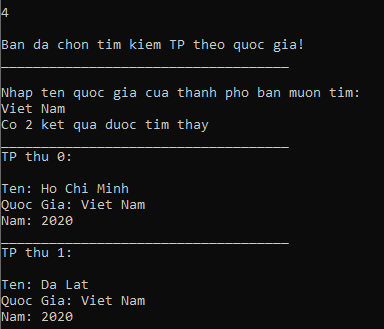
Hình 3.4

Chọn số 3: Tương tự như chức năng số 2 nhưng lần này chương trình sẽ hiển thị theo trình tự từ Z tới A.



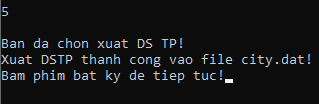
Hình 3.5

Chọn số 4: Chức năng này cho phép chúng ta nhập tên quốc gia và tìm kiếm các thành phố thuộc quốc gia đó và hiển thị ra màn hình console.



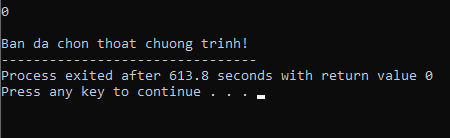
Hình 3.6

Chọn số 5: Chương trình sẽ ghi các dữ liệu đã nhập vào trong một tập tin nhị phân “city.dat”.



Hình 3.7

Chọn số 0: chương trình sẽ kết thúc và ấn một phím bất kỳ chương trình sẽ thoát.



Hình 3.8

**CHƯƠNG IV: TỔNG KẾT**

**Sau quá trình tìm hiểu và thực hiện đề tài đã được giao và nhóm F3 đã đạt được kết quả như sau:**

Hiểu rõ hơn về cách sử dụng các thuật toán

Trao dồi được kĩ năng mềm, kĩ năng làm việc nhóm, trao đổi tìm hiểu và đưa ra ý kiến của mỗi cá nhân.

Trang bị đầy đủ kiến thức để có thể vững bước trước những khó khăn kế tiếp.

**Những hạn chế của nhóm:**

Không thống nhất được thời gian cụ thể.

Chưa phát triển được giao diện cho chương trình.

**CHƯƠNG V: TƯ LIỆU THAM KHẢO**

<https://slideplayer.com/slide/14956438/>[truy cập 15/6/2020]

<https://www.youtube.com/watch?v=kfuv84LAl4c&list=PLE1qPKuGSJaBq4VFzTYrhzCiPvCoI8JDv&index=35>[truy cập 16/6/2020]

<https://www.youtube.com/watch?v=IMWCALHGxYk&list=PLE1qPKuGSJaBq4VFzTYrhzCiPvCoI8JDv&index=36>[truy cập 16/6/2020]

<https://www.youtube.com/watch?v=R06eV02zg6A&list=PLE1qPKuGSJaBq4VFzTYrhzCiPvCoI8JDv&index=34>[truy cập 16/6/2020]

[https://nguyenvanhieu.vn](https://nguyenvanhieu.vn/doc-ghi-file-trong-c/)[truy cập 16/6/2020]

<https://www.youtube.com/watch?v=r6XD86awquI&list=PLE1qPKuGSJaBq4VFzTYrhzCiPvCoI8JDv&index=38>[truy cập 16/6/2020]

<https://www.youtube.com/watch?v=G8WQPxh1PpM&t=129s>[truy cập 16/6/2020]

<https://www.slideshare.net/trongthuy3/luan-van-quan-li-xay-dung-theo-quy-hoach-khu-do-thi-moi-hay>[truy cập 25/6/2020]