BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP CƠ SỞ**

**Giảng viên hướng dẫn : Nguyễn Thủy Đoan Trang**

**Sinh viên thực hiện : Huỳnh Văn Trọng**

**Mã số sinh viên : 60137336**

Khánh Hòa: 2021

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------o0o---------------

****

**ĐỒ ÁN THỰC TẬP CƠ SỞ**

**TÊN ĐỀ TÀI:**

**CÀI ĐẶT CÁC THUẬT TOÁN TÌM KIẾM VÀ SẮP XẾP TRÊN DANH SÁCH LIÊN KẾT THEO   
HƯỚNG ĐỐI TƯỢNG**

**GVHD:** ThS. Nguyễn Thủy Đoan Trang

**SVTH:** Huỳnh Văn Trọng

**MSSV:** 60137336

Khánh Hòa, tháng 01/2021

**NHIỆM VỤ CỦA ĐỀ TÀI**

1. **Nội dung và các yêu cầu cần giải quyết trong đề tài thực tập cơ sở:**
2. **Nội dung:**

* Tìm kiếm, thu thập và tổng kết kiến thức liên quan đến đề tài thực tập sơ sở.
* Dựa theo bố cục đã được yêu cầu trong đề tài, phát triển đề tài đó ( xây dựng thuật toán, hiểu thật toán, code, tổng hợp kết quả nhận được ).
* Lập trình thử nghiệm đề tài thực tập cơ sở.

1. **Các yêu cầu cần phải giải quyết:**

* Mô tả đúng và đầy đủ kiến thức khi làm đề tài thực tập cơ sở cũng như các yêu cầu mà bài toán đề ra.
* Hoàn thành bản đồ án dựa theo các nội dung kể trên và tuân thủ theo đúng quy định của giáo viên hướng dẫn và nhà trường đã đề ra.
* Viết báo cáo tổng hợp, hoàn thành đề tài thực tập cơ sở.

**LỜI CẢM ƠN**

Trước hết, em xin bày tỏ tình cảm và lòng biết ơn của em tới cô giáo ThS.Nguyễn Thủy Đoan Trang. Người đã từng bước hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài thực tập cơ sở của mình.

Em xin chân thành cảm ơn các thầy cô giáo khoa công nghệ thông tin của trường Đại Học Nha Trang đã dìu dắt, dạy dỗ em cả về kiến thức chuyên môn và tinh thần học tập để em có được những kiến thức thực hiện đề tài thực tập cơ sở của mình.

Em xin chân thành cảm ơn **PGS. TS.Trang Sĩ Trung** – Hiệu trưởng Trường Đại HọcNha Trang, ban giám hiệu nhà trường, các phòng ban đã giúp đỡ tạo điều kiện tốt nhất cho em trong suốt thời gian học tập tại trường.

Tuy có nhiều cố gắng trong quá trình học tập, cũng như trong quá trình làm đề tài thực tập cơ sở không thể tránh khỏi những thiếu sót, em rất mong đƣợc sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn.

Khánh Hòa, ngày…tháng…năm….

Sinh viên thực hiện

Huỳnh Văn Trọng

**PHẦN MỞ ĐẦU**

1. **Lý do chọn đề tài:**

Công nghệ thông tin là một trong những ngành phát triển vượt bậc trong những năm gần đây. Ngày nay, với sự phát triển nhanh chóng của xã hội thì công nghệ thông tin được ứng dụng rỗng rãi ở hầu hết tất cả các lĩnh vực và ngày càng đóng vai trò quan trọng, trở thành một phần thiết yếu trong đời sống hằng ngày. Công nghệ thông tin là một ngành đòi hỏi ở người học phải hết sức kiên trì, có một nền tảng kiến thức vững chắc, sự tư duy logic cao, hiểu biết sâu rộng trên nhiều lĩnh vực. Chúng em hiện đang là sinh viên ngành công nghệ thông tin cần phải có sự đầu tư, không ngừng học hỏi để nâng cao kiến thức. Do đó để củng cố lại kiến thức đã học, đề tài thực tập cơ sở để em thực hiện là: **“ Cài đặt các thuật toán tìm kiếm và sắp xếp trên danh sách liên kết theo hướng đối tượng.”**

1. **Mục tiêu của đề tài**

* Củng cố lại kiến thức đã học về danh sách liên kết trong môn cấu trúc dữ liệu và giải thuật, vận dụng kiến thức đã được học từ môn “Lập trình hướng đối tượng”.
* Rèn luyện kĩ năng lập trình trên ngôn ngữ C++.
* Ứng dụng các lý thuyết đã học để giải quyết bài toán ứng dụng cụ thể.
* Ứng dụng phương pháp lập trình hướng đối tượng để thực hiện các thuật toán cấu trúc dữ liệu.

1. **Phạm vi nghiên cứu:**

* Lý thuyết về danh sách liên kết trong môn cấu trúc dữ liệu và giải thuật.
* Lý thuyết của môn “Lập trình hướng đối tượng”.

1. **Phương pháp nghiên cứu:**

* Nghiên cứu tài liệu “Cấu trúc dữ liệu và giải thuật” và “Lập trình hướng đối tượng”.
* Tìm kiếm và nghiên cứu trên mạng Internet.

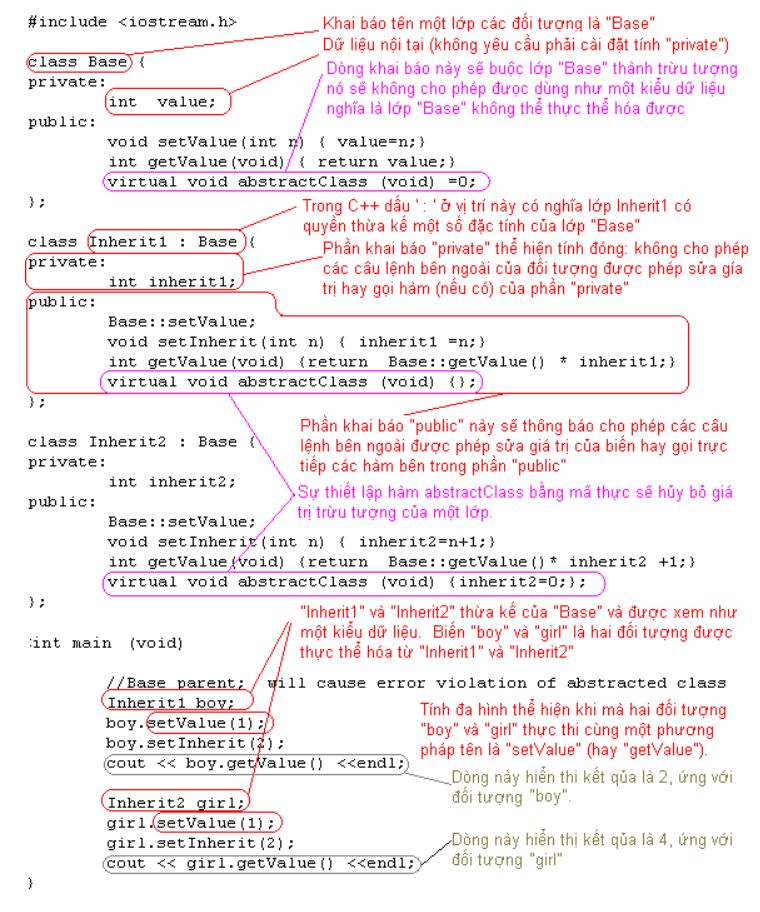
1. **Các khái niệm liên quan đến đề tài:**
2. **Lập trình hướng đối tượng là gì ?**

Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt: OOP) là một mẫu hình lập trình dựa trên khái niệm "công nghệ đối tượng", mà trong đó, đối tượng chứa đựng các dữ liệu, trên các trường, thường được gọi là các thuộc tính; và mã nguồn, được tổ chức thành các phương thức. Phương thức giúp cho đối tượng có thể truy xuất và hiệu chỉnh các trường dữ liệu của đối tượng khác, mà đối tượng hiện tại có tương tác (đối tượng được hỗ trợ các phương thức "this" hoặc "self").

Lập trình hướng đối tượng giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, nhiều người cho rằng OOP dễ tiếp thu hơn và giảm nhẹ các thao tác viết mã cho người lập trình.

Các ngôn ngữ lập trình hướng đối tượng đáng chú ý gồm có Java, C++, C#, Python, PHP, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, và Smalltalk.

*( Nguồn: Wikipedia )*



*Hình 1: Lập trình hướng đối tượng bằng C++*

*( Nguồn: Wikipedia )*

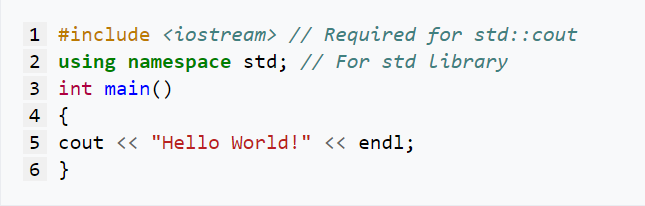
1. **Ngôn ngữ C++ là gì ?**

Ngôn ngữ C++ là một loại ngôn ngữ lập trình bậc trung. Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại ưa thích và phổ biến của lập trình viên.

Ngôn ngữ C++ được thiết kế hướng tới lập trình hệ thống máy tính và phần mềm nhúng trên các mạch vi xử lý, bao gồm cả hệ thống có tài nguyên hạn chế và tài nguyên khổng lồ, với ưu điểm vượt trội về hiệu suất, hiệu quả và tính linh hoạt cao.

*( Nguồn: Wikipedia )*

**Ví dụ đơn giản:** dùng hàm thư viện chuẩn C++ “cout” để hiển thị một thông báo và kết thúc chương trình.



*Hình 2: Ví dụ đơn giản của ngôn ngữ C++*

1. **Danh sách liên kết là gì ?**
2. **Khái niệm danh sách liên kết:**

Danh sách liên kết là một cấu trúc dữ liệu tuyến tính, trong đó các phần tử không được lưu trữ tại các vị trí bộ nhớ liền kề. Các phần tử trong danh sách được liên kết được liên kết bằng cách sử dụng con trỏ như được hiển thị trong hình ảnh bên dưới:

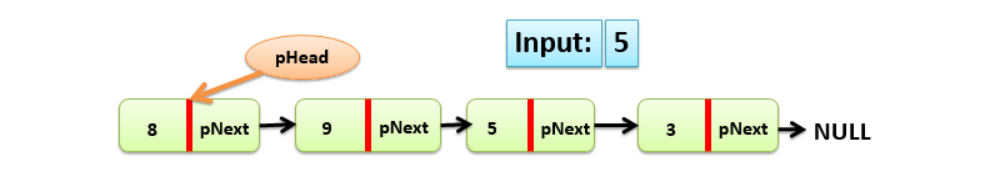


*Hình 3: Ví dụ danh sách liên kết*

1. **Tìm kiếm phần tử trong danh sách liên kết:**

Trong phần này mình sẽ thực hiện tìm kiếm một giá trị index được nhập từ phím trong danh sách liên kết đơn. Đây là một thao tác đơn giản, chỉ cần ta duyệt từng phần tử trong danh sách.

**Ví dụ:**



*Hình 4: Ví dụ tìm kiếm phần tử trong danh sách liên kết*

Giả sử chúng ta có giá trị index = 5 là giá trị cần tìm trong danh sách. Lúc này ta cần khai báo một Node tạm pTmp để thay thế cho pHead duyệt danh sách.

Thực hiện vòng lặp while lặp từng phần tử trong danh sách với điều kiện pTmp != Null. Nếu pTmp => data = index thì thoát khỏi vòng lặp và thông báo đã tìm thấy, ngược lại thì thông báo không.

1. **Sắp xếp phần tử trong danh sách liên kết:**

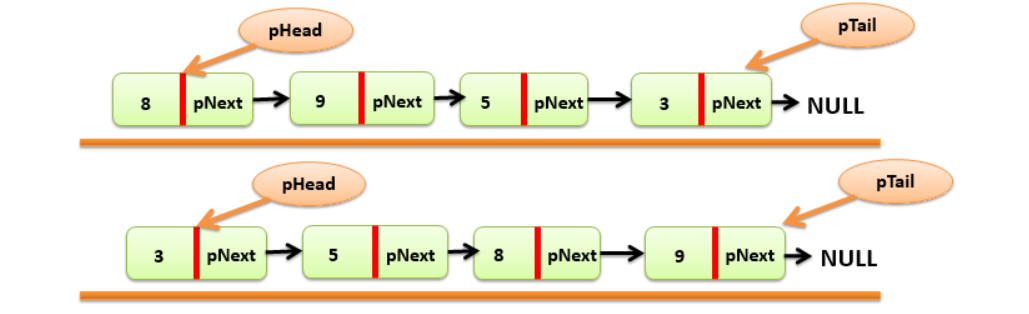
Thuật toán sắp xếp là lời giải của bài toán sắp xếp, vậy thì trước tiên, ta hãy tìm hiểu xem bài toán sắp xếp là gì trước đã.

Bài toán sắp xếp chắc chắn không còn xa lạ gì với mỗi chúng ta, nó là một trong những bài toán được bắt gặp nhiều nhất trong thực tế.

Bài toán sắp xếp là chúng ta sẽ sắp xếp lại các phần tử của một danh sách theo chiều tăng hoặc giảm dần theo một tiêu chí nào đó của phần tử trong danh sách.

Trong lập trình, sắp xếp không chỉ đơn giản là để tìm một hoặc nhiều phần tử đứng đầu về một tiêu chí nào đó hay để có cái nhìn tổng quan về dữ liệu, sắp xếp còn làm cơ sở cho các giải thuật nâng cao với hiệu suất cao hơn.

**Ví dụ:**



*Hình 5: Ví dụ sắp xếp phần tử trong danh sách liên kết*

**c.1. Thuật toán Buble-sort (nổi bọt):**

Sắp xếp nổi bọt hay bubble sort là thuật toán sắp xếp đầu tiên và cũng là thuật toán đơn giản nhất trong các thuật toán sắp xếp khác, ý tưởng của thuật toán này như sau:

Duyệt qua danh sách, làm cho các phần tử lớn nhất hoặc nhỏ nhất dịch chuyển về phía cuối danh sách, tiếp tục lại làm phần tử lớn nhất hoặc nhỏ nhất kế đó dịch chuyển về cuối hay chính là làm cho phần tử nhỏ nhất (hoặc lớn nhất) nổi lên, cứ như vậy cho đến hết danh sách cụ thể các bước thực hiện của giải thuật này như sau:

- Bước 1: Gán i=0

- Bước 2: Gán j=0

- Bước 3: Nếu A[j] > A[j + 1] thì đối chỗ A[j] và A[j + 1]

- Bước 4: Nếu j < n – i – 1:

+ Đúng thì j = j + 1 và quay lại bước 3

+ Sai thì sang bước 5

- Bước 5: Nếu i < n – 1:

+ Đúng thì i = i + 1 và quay lại bước 2

+ Sai thì dừng lại

**c.2. Thuật toán Selection-sort (chọn):**

Sắp xếp chọn hay selection sort sẽ là thuật toán thứ hai , ý tưởng của thuật toán này như sau: duyệt từ đầu đến phần tử kề cuối danh sách, duyệt tìm phần tử nhỏ nhất từ vị trí kế phần tử đang duyệt đến hết, sau đó đổi vị trí của phần tử nhỏ nhất đó với phần tử đang duyệt và cứ tiếp tục như vậy.

Cho mảng A có n phần tử chưa được sắp xếp. Cụ thể các bước của giải thuật này áp dụng trên mảng A như sau:

* Bước 1: Gán i = 0
* Bước 2: Gán j = i + 1 và min = A[i]
* Bước 3: Nếu j < n:

+ Nếu A[j] < A[min] thì min = j

+ j = j + 1

+ Quay lại bước 3

* Bước 4: Đổi chỗ A[min] và A[i]
* Bước 5: Nếu i < n – 1:

+ Đúng thì i = i + 1 và quay lại bước 2

+ Sai thì dừng lại

**c.3. Thuật toán Insert-sort (chèn):**

Sắp xếp chèn hay insertion sort là thuật toán tiếp theo, ý tưởng của thuật toán này như sau: ta có mảng ban đầu gồm phần tử A[0] xem như đã sắp xếp, ta sẽ duyệt từ phần tử 1 đến n – 1, tìm cách chèn những phần tử đó vào vị trí thích hợp trong mảng ban đầu đã được sắp xếp.

Giả sử cho mảng A có n phần tử chưa được sắp xếp. Các bước thực hiện của thuật toán áp dụng trên mảng A như sau:

* Bước 1: Gán i = 1
* Bước 2: Gán x = A[i] và pos = i – 1
* Bước 3: Nếu pos >= 0 và A[pos] > x:

+ A[pos + 1] = A[pos]

+ pos = pos – 1

+ Quay lại bước 3

* Bước 4: A[pos + 1] = x
* Bước 5: Nếu i < n:

+ Đúng thì i = i + 1 và quay lại bước 2

+ Sai thì dừng lại

**c.4. Thuật toán Quicksort:**

Sắp xếp nhanh (quick sort) hay sắp xếp phân đoạn (Partition) là thuật toán sắp xếp dựa trên kỹ thuật chia để trị, cụ thể ý tưởng là: chọn một điểm làm chốt (gọi là pivot), sắp xếp mọi phần tử bên trái chốt đều nhỏ hơn chốt và mọi phần tử bên phải đều lớn hơn chốt, sau khi xong ta được 2 dãy con bên trái và bên phải, áp dụng tương tự cách sắp xếp này cho 2 dãy con vừa tìm được cho đến khi dãy con chỉ còn 1 phần tử.

Cụ thể áp dụng thuật toán cho mảng như sau:

1. Chọn một phần tử làm chốt

2. Sắp xếp phần tử bên trái nhỏ hơn chốt

3. Sắp xếp phần tử bên phải nhỏ hơn chốt

4. Sắp xếp hai mảng con bên trái và bên phải pivot

Phần tử được chọn làm chốt rất quan trọng, nó quyết định thời gian thực thi của thuật toán. Phần tử được chọn làm chốt tối ưu nhất là phần tử trung vị, phần tử này làm cho số phần tử nhỏ hơn trong dãy bằng hoặc sấp xỉ số phần tử lớn hơn trong dãy. Tuy nhiên, việc tìm phần tử này rất tốn kém, phải có thuật toán tìm riêng, từ đó làm giảm hiệu suất của thuật toán tìm kiếm nhanh, do đó, để đơn giản, người ta thường sử dụng phần tử chính giữa làm chốt.

**c.5. Thuật toán Heap-sort ( vun đống ):**

Sắp xếp vun đống (Heap Sort) là một kỹ thuật sắp xếp phân loại dựa trên một cấu trúc dữ liệu được gọi là đống nhị phân (binary heap), gọi đơn giản là đống. Nó tương tự như thuật toán Sắp xếp chọn (Selection Sort) nơi phần tử lớn nhất sẽ được xếp vào cuối danh sách. Lặp đi lặp lại các bước này cho các phần tử còn lại của danh sách.

**c.6. Thuật toán Merge Sort ( sắp xếp trộn ):**

Sắp xếp trộn (Merge Sort) là một thuật toán dựa trên kỹ thuật chia để trị, ý tưởng của thuật toán này như sau: chia đôi mảng thành hai mảng con, sắp xếp hai mảng con đó và trộn lại theo đúng thứ tự, mảng con được sắp xếp bằng cách tương tự.

Giả sử left là vị trí đầu và right là cuối mảng đang xét, cụ thể các bước của thuật toán như sau:

Nếu mảng còn có thể chia đôi được (tức left < right)

1. Tìm vị trí chính giữa mảng

2. Sắp xếp mảng thứ nhất (từ vị trí left đến mid)

3. Sắp xếp mảng thứ 2 (từ vị trí mid + 1 đến right)

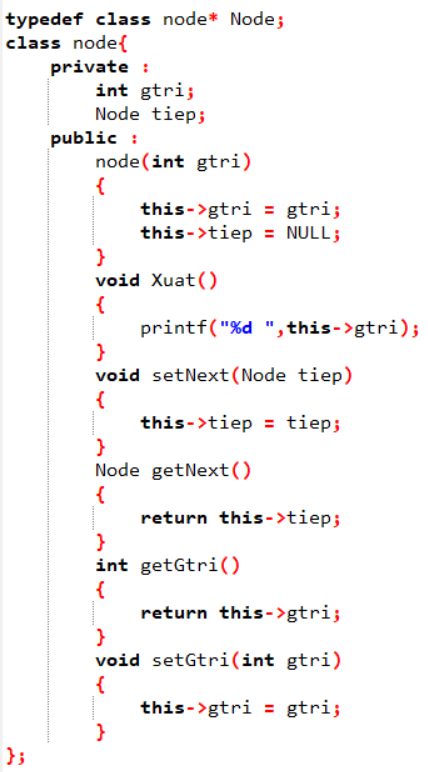
4. Trộn hai mảng đã sắp xếp với nhau

1. **Giải thích thuật toán ở mức 1:**
   1. **Thuật toán nhập giá trị từ bàn phím là lấy giá trị ngẫu nhiên:**

* Bước 1:

+ Tạo đối tượng Node có các thuộc tính **gtri** và **tiep**

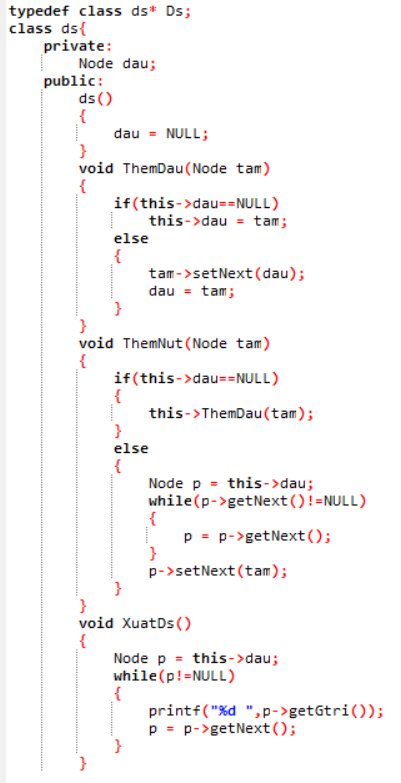
+ Xây dựng các phương thức tạo đối tượng mới như: **Xuat(), setNext(), getNext(), getGtri, setGtri().**



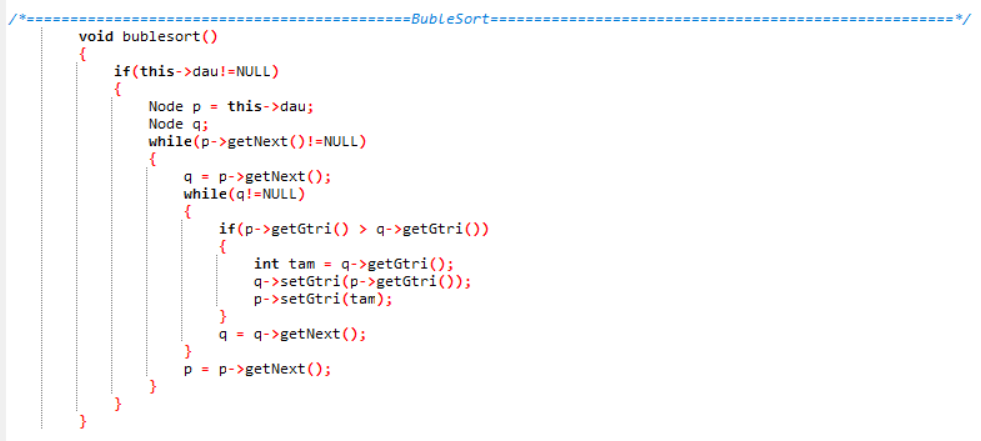
* Bước 2:

+ Tạo đối tượng **Ds** có các biến khai báo là các thuộc tính.

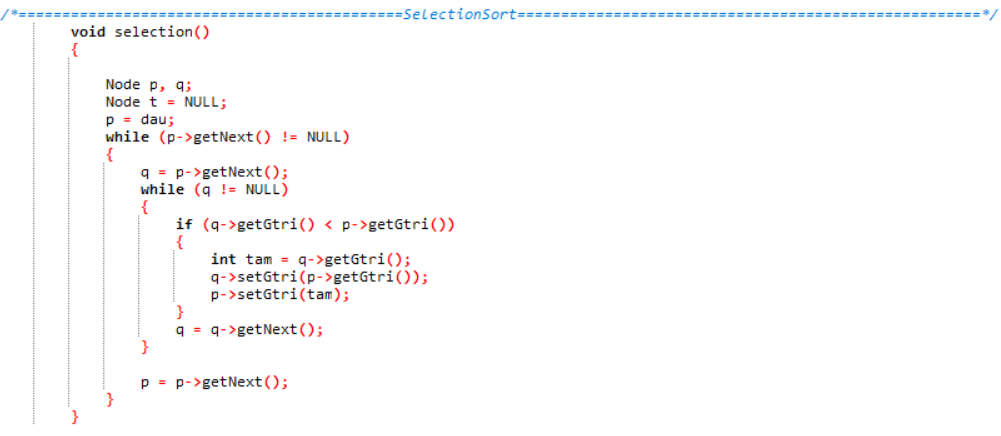
+ Xây dựng các hàm là phương thức của các đối tượng.



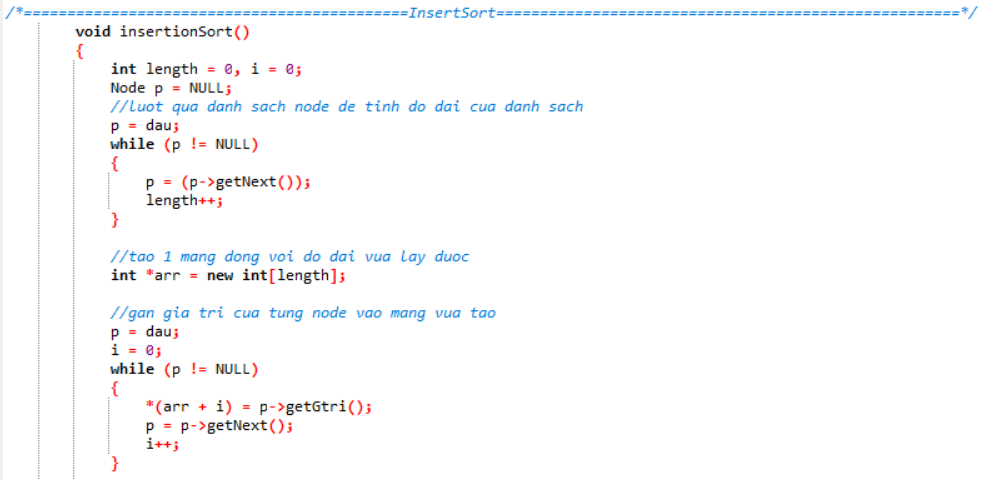
* Bước 3: Xây dựng thuật toán sắp xếp **Buble Sort** tăng dần:

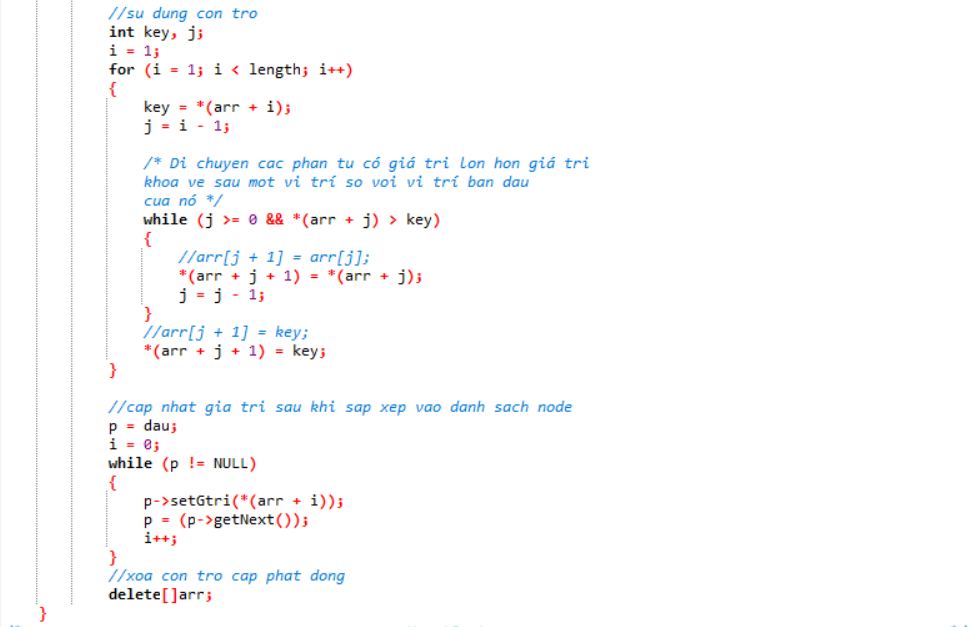


* Bước 4: Xây dựng thuật toán sắp xếp **Selection Sort** tăng dần:

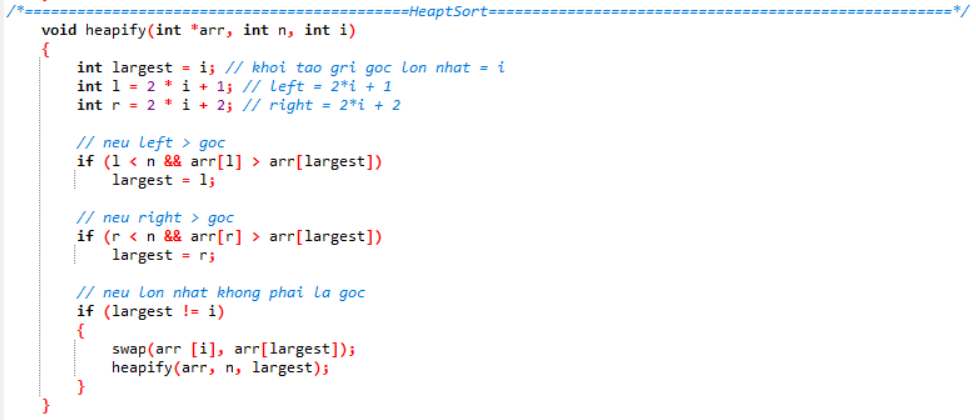


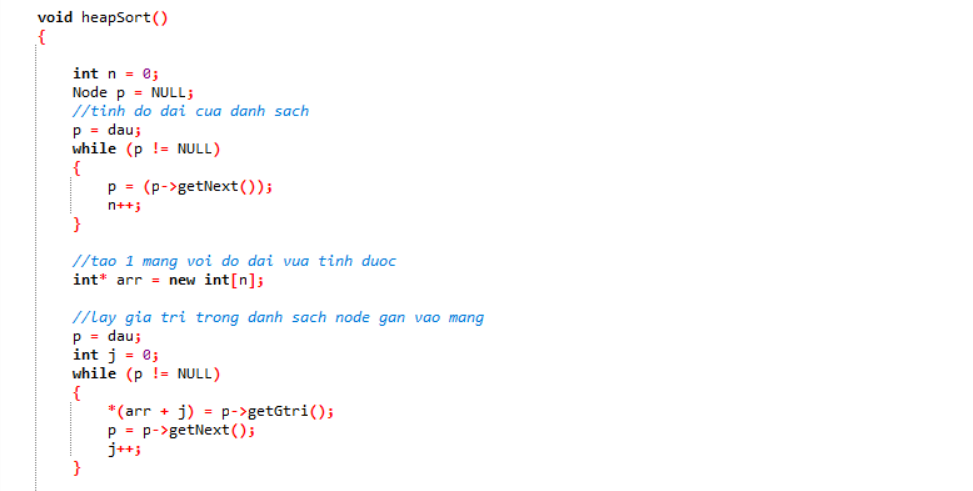
* Bước 5: Xây dựng thuật toán sắp xếp **Insertion Sort** tăng dần:

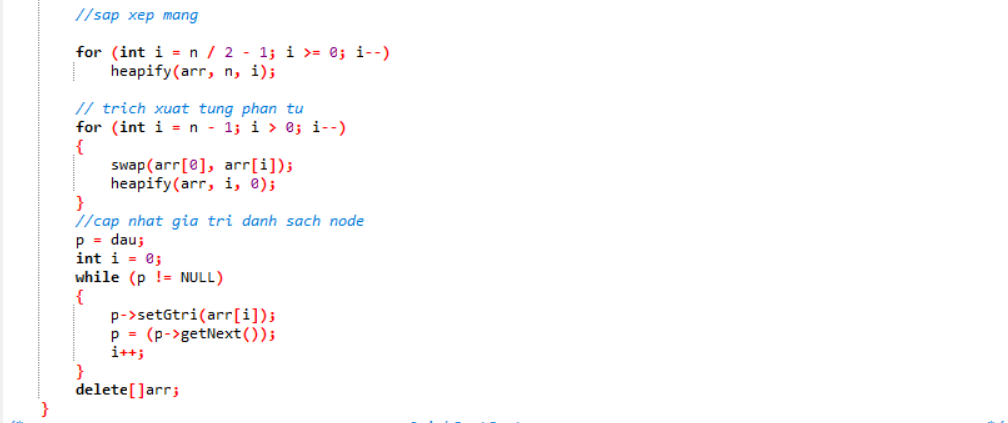




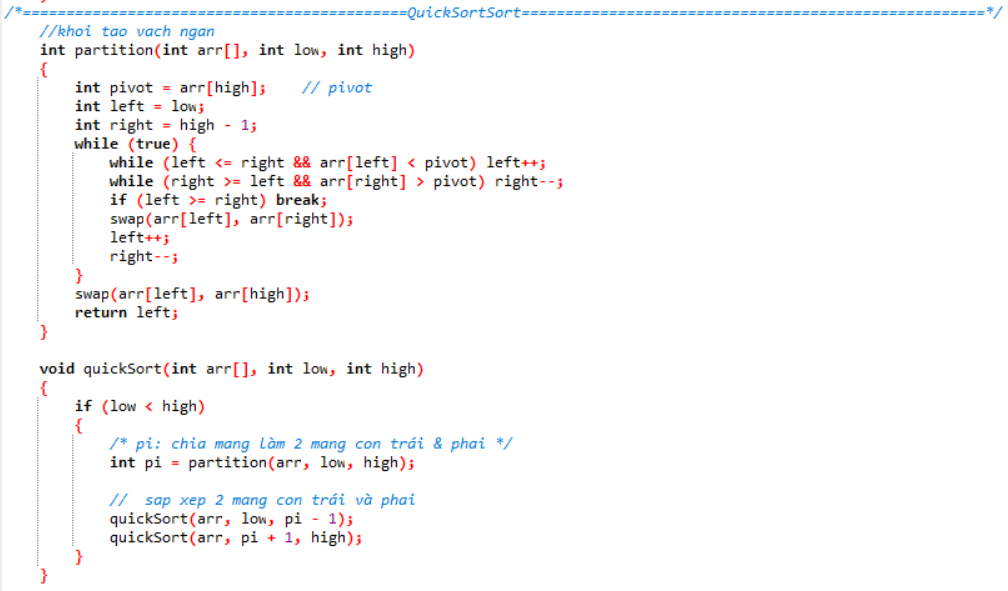
* Bước 6: Xây dựng thuật toán sắp xếp **Heap Sort** tăng dần:

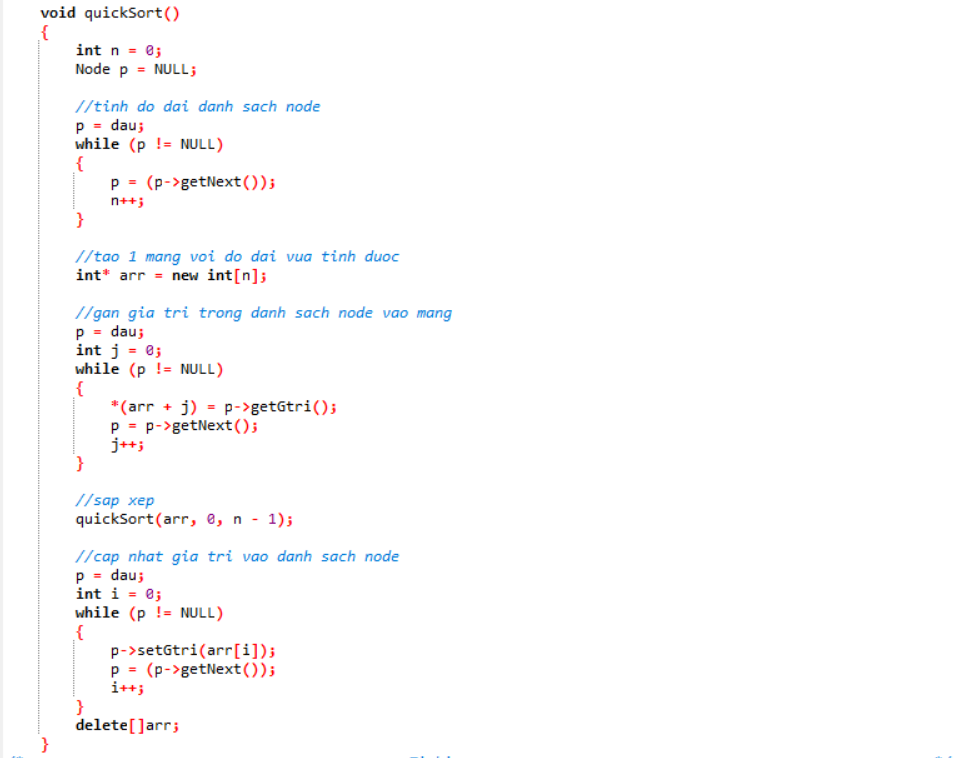




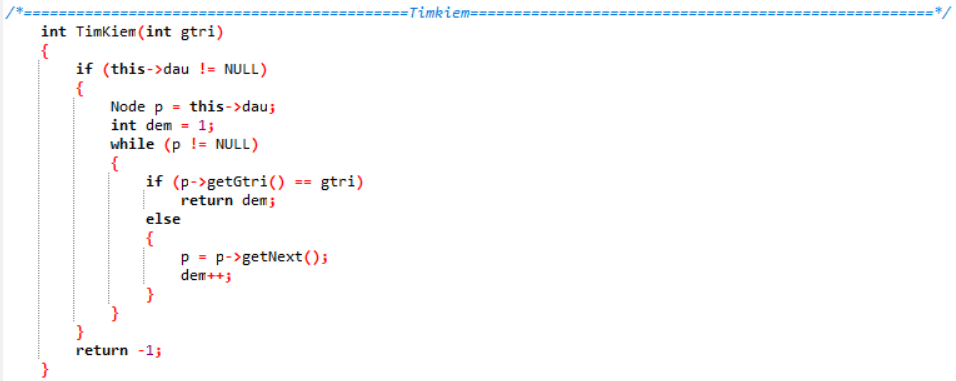


* Bước 7:





* Bước 8: Tìm kiếm phần tử trong danh sách liên kết.



* 1. **Thuật toán lấy giá trị từ FILE:**

Giống hầu hết tất cả như các bước của thuật toán nhập giá trị từ bàn phím là lấy giá trị ngẫu nhiên. Nhưng có thêm bước 9: **Đọc file**

