**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO THỰC TẬP CƠ SỞ**

**ĐỀ TÀI:**

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

**GVHD: Nguyễn Thủy Đoan Trang**

**SVTH: Trần Huy Thịnh**

**MSSV: 57131383**

**Lớp: 59CNTT1**

**NỘI DUNG PHẢI NỘP TRONG BÁO CÁO KHI KẾT THÚC THỰC TẬP**

**MỤC LỤC**

[**CHƯƠNG 1.** **THÔNG TIN ĐỀ TÀI** 4](#_Toc30167651)

[**1.** **Tên đề tài:** 4](#_Toc30167652)

[**2.** **Thông tin sinh viên:** 4](#_Toc30167653)

[**3.** **Thông tin giảng viên hướng dẫn:** 4](#_Toc30167654)

[**4.** **Thời gian thực hiện:** 4](#_Toc30167658)

[**CHƯƠNG 2.** **TÓM TẮT ĐỀ TÀI:** 5](#_Toc30167660)

[**1.** **Nội dung:** 5](#_Toc30167661)

[**2.** **Công cụ:** 5](#_Toc30167662)

[**CHƯƠNG 3.** **CƠ SỞ LÝ THUYẾT:** 6](#_Toc30167665)

[**1.** **Danh sách:** 6](#_Toc30167666)

[**3.** **Danh sách liên kết:** 6](#_Toc30167667)

[**4.** **Nút của danh sách liên kết:** 6](#_Toc30167668)

[**5.** **Các thuật toán tìm kiếm:** 7](#_Toc30167669)

[**5.1.** **Tìm kiếm tuần tự:** 7](#_Toc30167670)

[**5.2.** **Tìm kiếm nhị phân (Binary search):** 7](#_Toc30167681)

[**6.** **Các thuật toán sắp xếp:** 8](#_Toc30167682)

[**6.1.** **Sắp xếp nổi bọt (Bubble sort):** 8](#_Toc30167683)

[**6.2.** **Sắp xếp chọn (Selection sort):** 8](#_Toc30167689)

[**6.3.** **Sắp xếp chèn (Insertion sort):** 8](#_Toc30167690)

[**6.4.** **Sắp xếp chất đống (Heap sort):** 8](#_Toc30167691)

[**6.5.** **Sắp xếp nhanh hay sắp xếp phân đoạn (Quick sort / Partition sort):** 9](#_Toc30167700)

[**6.6.** **Sắp xếp trộn trực tiếp (Straight merge sort):** 9](#_Toc30167701)

[**CHƯƠNG 4.** **KẾ HOẠCH THỰC HIỆN:** 10](#_Toc30167704)

[**CHƯƠNG 5.** **NỘI DUNG THỰC HIỆN:** 11](#_Toc30167705)

[**1.** **Sơ đồ lớp:** 11](#_Toc30167706)

[**2.** **Các phương thức cơ bản:** 12](#_Toc30167707)

[**2.1.** **setNode():** Thay đổi các thành phần của nút. 12](#_Toc30167708)

[**2.2.** **linkNode():** Gắn nút vào danh sách. 12](#_Toc30167709)

[**2.3.** **createList():** Chọn phương pháp tạo và tạo danh sách. 12](#_Toc30167710)

[**2.4.** **sortList():** Chọn phương pháp sắp xếp danh sách. 13](#_Toc30167711)

[**2.5.** **getList():** In danh sách ra màn hình. 14](#_Toc30167712)

[**2.6.** **getLast():** Tìm nút cuối danh sách. 14](#_Toc30167713)

[**2.7.** **swapData():** Đảo giá trị hai nút. 14](#_Toc30167714)

[**2.8.** **writeList():** Ghi danh sách vào file kết quả. 15](#_Toc30167715)

[**2.9.** **partition():** Tạo phân đoạn cho sắp xếp nhanh. 15](#_Toc30167716)

[**2.10.** **halveList():** Chia đôi đoạn danh sách. 16](#_Toc30167728)

[**2.11.** **mergeLists():** Trộn hai đoạn danh sách. 16](#_Toc30167729)

[**3.** **Các phương thức nhập, xuất dữ liệu:** 17](#_Toc30167730)

[**3.1.** **keyList():** Nhập dữ liệu từ bàn phím. 17](#_Toc30167731)

[**3.2.** **randList():** Sinh dữ liệu ngẫu nhiên. 17](#_Toc30167732)

[**3.3.** **fileList():** Nhập dữ liệu từ file. 18](#_Toc30167733)

[**4.** **Phương thức tìm kiếm:** 18](#_Toc30167734)

[**5.** **Các phương thức sắp xếp:** 19](#_Toc30167735)

[**5.1.** **bubbleSort():** 19](#_Toc30167736)

[**5.2.** **selectionSort():** 20](#_Toc30167737)

[**5.3.** **insertionSort():** 21](#_Toc30167738)

[**5.4.** **quickSort():** 22](#_Toc30167739)

[**5.5.** **mergeSort():** 23](#_Toc30167740)

[**6.** **Chương trình chính:** 24](#_Toc30167741)

[**7.** **Thư viện sử dụng:** 24](#_Toc30167742)

[**7.1.** **Nguồn:** 24](#_Toc30167743)

[**7.2.** **Các thư viện dựng sẵn:** 24](#_Toc30167744)

[**7.3.** **Các macro:** 24](#_Toc30167745)

[**7.4.** **Các hằng số :** 25](#_Toc30167746)

[**7.5.** **Các hàm được sử dụng trong thư viện:** 25](#_Toc30167747)

[**CHƯƠNG 6.** **KẾT QUẢ THỰC HIỆN** 27](#_Toc30167748)

[**1.** **Kết quả đạt được:** 27](#_Toc30167749)

[**2.** **Một số hình ảnh:** 28](#_Toc30167750)

[**CHƯƠNG 7.** **TÀI LIỆU THAM KHẢO:** 30](#_Toc30167751)

1. **THÔNG TIN ĐỀ TÀI**
   1. **Tên đề tài:**

**Cài đặt các thuật toán tìm kiếm và sắp xếp trên danh sách liên kết theo hướng đối tượng**

* 1. **Thông tin sinh viên:**

**Họ Tên:** Trần Huy Thịnh **MSSV:** 57131383

**Khóa:** 59 **Lớp:** 59CNTT1

**Email:** tranhuythinh97@gmail..com

* 1. **Thông tin giảng viên hướng dẫn:**

**Giảng viên:** [Nguyễn Thủy Đoan Trang](https://elearning.ntu.edu.vn/user/view.php?id=7&course=1077)

**Email:** nguyenthuydoantrang@ntu.edu.vn

4. **Thời gian thực hiện:**

Từ ngày 13/12/2019 đến ngày 17/01/2020

2. **TÓM TẮT ĐỀ TÀI:**
   1. **Nội dung:**

* Đề tài thực hiện tìm hiểu và cài đặt các thuật toán tìm kiếm, sắp xếp trên danh sách liên kết đơn gồm các số nguyên.
* Thuật toán tìm kiếm trên danh sách liên kết: tìm kiếm tuần tự.
* Các thuật toán sắp xếp trên danh sách liên kết: sắp xếp nổi bọt (Bubble sort), sắp xếp chọn(Selection sort), sắp xếp chèn (Insertion sort), sắp xếp nhanh (Quick sort), sắp xếp trộn (Merge sort).
* Danh sách liên kết chỉ có thể truy xuất tuần tự nên việc thực hiện các phương thức tìm kiếm nhị phân (Binary search), sắp xếp chất đống (Heap sort) không hiệu quả vì các phương thức này phụ thuộc chủ yếu vào truy xuất ngẫu nhiên.
  1. **Công cụ:**
* Phần mềm Dev C++ 5.11.
* Ngôn ngữ C++.
* Các phương thức nhập dữ liệu: từ bàn phím, từ file, chọn số ngẫu nhiên.
* Các phương thức xuất dữ liệu: in kết quả lên màn hình, ghi vào file.

3. **CƠ SỞ LÝ THUYẾT:**
4. **Danh sách:**

Danh sách là một tập hợp hữu hạn các phần tử. Giữa các phần tử có mối liên hệ tương đối: nếu biết vị trí của phần tử thứ i thì sẽ xác định được vị trí của phần tử thứ i+1.

1. **Danh sách đặc:**

Danh sách đặc là một danh sách có các phần tử được lưu ở các vị trí kế tiếp nhau trong bộ nhớ như mảng (array), vector, stack, queue.

1. **Danh sách liên kết:**

Danh sách liên kết là một danh sách có vị trí các phần tử được quản lí bằng con trỏ địa chỉ, các phần tử trong danh sách được liên kết với nhau bằng các con trỏ địa chỉ này. Các phần tử trong danh sách liên kết không nhất thiết phải được lưu ở các vị trí kế tiếp nhau trong bộ nhớ. Mỗi phần tử của danh sách liên kết được gọi là một nút của danh sách.

1. **Nút của danh sách liên kết:**

Mỗi nút của danh sách liên kết gồm hai thành phần:

Thông tin của nút (Info): chứa giá trị hay các thông tin của nút đó.

Liên kết của nút (Link): các con trỏ chứa địa chỉ của các nút liên quan.

Đối với danh sách liên kết đơn, mỗi nút chỉ có một liên kết là con trỏ chứa địa chỉ của nút tiếp theo.

1. **Các thuật toán tìm kiếm:**
   1. **Tìm kiếm tuần tự:**

Thuật toán này thực hiện tìm kiếm một phần tử trong danh sách bằng cách duyệt lần lượt bắt đầu từ phần tử đầu hoặc cuối danh sách, đi theo hướng ngược lại đến khi tìm thấy phần tử cần tìm hoặc khi đã duyệt hết các phần tử trong danh sách.

8. 2. **Tìm kiếm nhị phân (Binary search):**

Thuật toán này thực hiện tìm kiếm một phần tử trong danh sách bằng cách truy xuất đến phần tử nằm chính giữa danh sách. Nếu danh sách đã được sắp theo thứ tự tăng dần, thì nếu phần tử nằm chính giữa danh sách có giá trị nhỏ hơn giá trị cần tìm thì tiếp tục truy xuất đến phần tử nằm chính giữa của nửa trước danh sách, ngược lại thì truy xuất đến phần tử nằm chính giữa của nửa sau danh sách, quá trình này lặp lại đến khi tìm thấy phần tử cần tìm hoặc khi đã truy xuất hết một nửa số phần tử trong danh sách.

Thuật toán dựa trên việc so sánh hơn kém giá trị các phần tử với giá trị cần tìm nên chỉ áp dụng được sau khi đã sắp xếp danh sách theo thứ tự tăng dần hoặc giảm dần.

Ngoài ra, hầu hết các bước của thuật toán chỉ truy xuất đến các phần tử nằm giữa danh sách, mà trong danh sách liên kết chỉ có thể duyệt tuần tự từ một đầu danh sách, nên việc truy xuất đến các phần tử nằm giữa danh sách đối với danh sách liên kết phải thông qua duyệt tuần tự nên thuật toán này sẽ mất nhiều thời gian thực hiện trên danh sách liên kết hơn so với trên mảng.

Vì vậy, đề tài không áp dụng thuật toán này.

1. **Các thuật toán sắp xếp:**
   1. **Sắp xếp nổi bọt (Bubble sort):**

Thuật toán này thực hiện sắp xếp danh sách bằng cách duyệt và so sánh các phần tử với nhau từng đôi một và đảo vị trí nếu hai phần tử không thỏa yêu cầu sắp xếp.

Thuật toán kết thúc khi tất cả các phần tử trong danh sách đã được so sánh với nhau.

4. 2. **Sắp xếp chọn (Selection sort):**

Thuật toán sắp xếp danh sách bằng cách duyệt danh sách từ đầu. Nếu yêu cầu sắp xếp tăng dần thì tìm phần tử nhỏ nhất trong đoạn còn lại, rồi so sánh với phần tử đang xét ở đầu đoạn danh sách và đảo vị trí nếu phần tử tìm được nhỏ hơn phần tử ở đầu đoạn danh sách.

Thuật toán kết thúc khi đoạn đang xét còn lại một phần tử.

* 1. **Sắp xếp chèn (Insertion sort):**

Thuật toán sắp xếp danh sách bằng cách tạo một danh sách mới, rồi lần lượt chèn các phần tử trong danh sách cũ vào vị trí thích hợp trong danh sách mới.

Thuật toán kết thúc khi tất cả các phần tử đã được chèn vào danh sách mới.

* 1. **Sắp xếp chất đống (Heap sort):**

Thuật toán sắp xếp danh sách bằng cách tổ chức lại các phần tử theo cấu trúc chất đống (Heap): nếu yêu cầu sắp tăng dần thì phần tử thứ i có giá trị lớn hơn giá trị các phần tử thứ i\*2 và i\*2+1.

Thuật toán dựa trên việc so sánh các phần tử không liền kề nhau và thao tác chủ yếu trên các phần tử nằm giữa danh sách. Ngoài ra, cấu trúc Heap có dạng gần giống với cấu trúc cây (Tree) nên thuật toán chỉ có hiệu quả khi áp dụng đối với mảng và cây.

Vì vậy, đề tài không áp dụng thuật toán này.

4. 5. **Sắp xếp nhanh hay sắp xếp phân đoạn (Quick sort / Partition sort):**

Thuật toán sắp xếp danh sách bằng cách chia danh sách thành các phân đoạn so với một phần tử nằm giữa danh sách. Nếu yêu cầu sắp xếp tăng dần thì phân đoạn bên trái gồm các phần tử có giá trị nhỏ hơn và phân đoạn bên phải gồm các phần tử có giá trị lớn hơn so với phần tử nằm giữa.

Quá trình sắp xếp được lặp lại với các phân đoạn đến khi mỗi phân đoạn chỉ còn một phần tử.

* 1. **Sắp xếp trộn trực tiếp (Straight merge sort):**

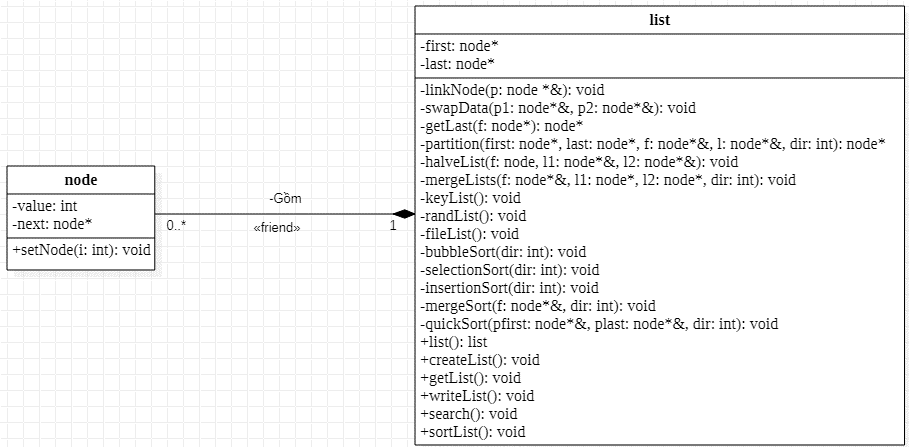
Thuật toán sắp xếp danh sách bằng cách chia nửa danh sách đến khi các đoạn danh sách còn một phần tử, sau đó lần lượt trộn các phân nửa đoạn lại với nhau theo thứ tự sắp xếp.

Thuật toán kết thúc khi đã trộn hết các đoạn với nhau.

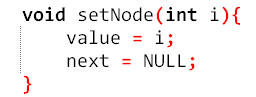
3. **KẾ HOẠCH THỰC HIỆN:**

|  |  |  |
| --- | --- | --- |
| **Tuần** | **Công việc dự kiến** | **Thực hiện** |
|  | Chọn đề tài. | **X** |
| Viết đề cương chi tiết và kế hoạch thực hiện. | **X** |
| Thiết kế các lớp đối tượng, cài đặt các phương thức nhập, xuất dữ liệu. | **X** |
| Cài đặt phương thức tìm kiếm tuần tự. | **X** |
| Cài đặt các phương thức Bubble Sort, Selection Sort. | **19/12-21/12** |
| Viết và nộp báo cáo quá trình thực hiện tuần 1. | **22/12-23/12** |
|  | Cài đặt phương thức, Insertion Sort. | **23/12-25/12** |
| Cài đặt phương thức Quick Sort. | **25/12-27/12** |
| Viết và nộp báo cáo quá trình thực hiện tuần 2. | **28/12-30/12** |
|  | Cài đặt phương thức Merge Sort. | **30/12-02/01** |
| Viết báo cáo quá trình thực hiện tuần 3. | **03/01-04/01** |
| Tổng kết và hoàn thành báo cáo đợt thực tập. | **04/01-05/01** |
|  | Kiểm tra lại chương trình và nội dung báo cáo, sửa lỗi nếu có. | **06/01-14/01** |

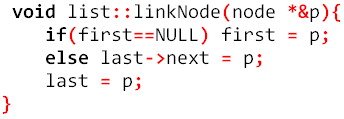
1. **NỘI DUNG THỰC HIỆN:**
2. **Sơ đồ lớp:**



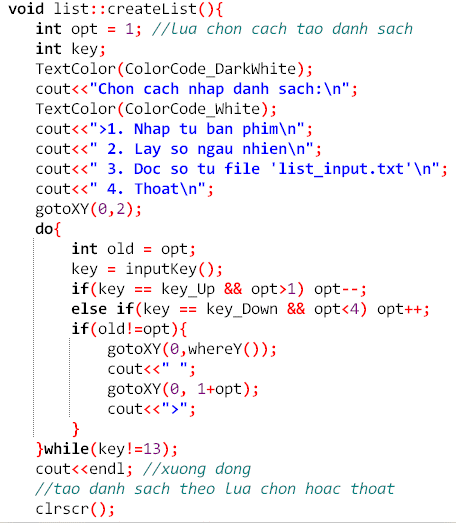
1. **Các phương thức cơ bản:**
   1. **setNode():** Thay đổi các thành phần của nút.

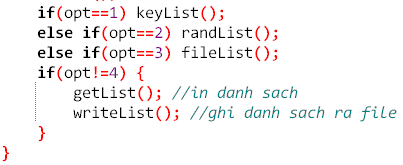


* 1. **linkNode():** Gắn nút vào danh sách.

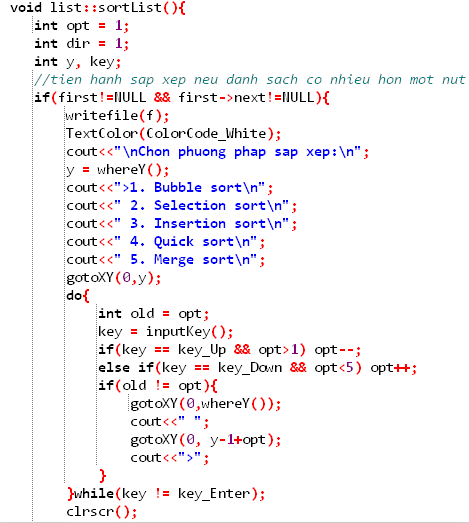


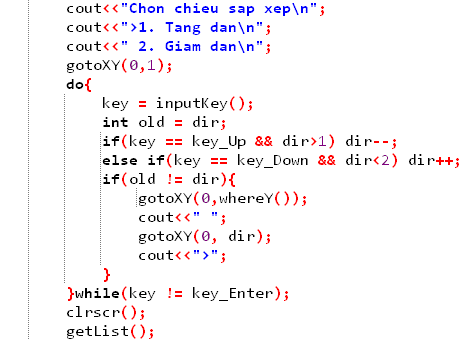
* 1. **createList():** Chọn phương pháp tạo và tạo danh sách.

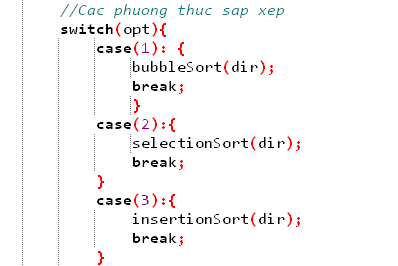


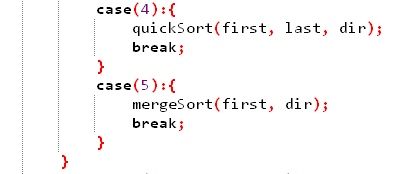


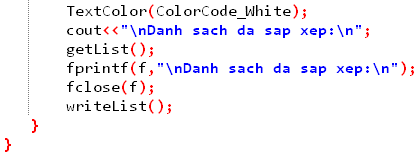
* 1. **sortList():** Chọn phương pháp sắp xếp danh sách.



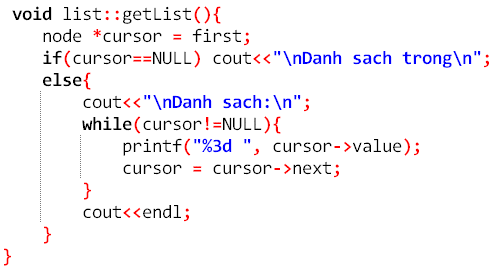




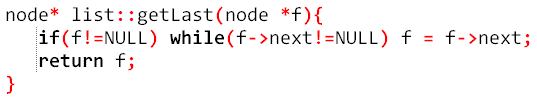




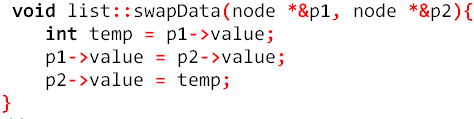
* 1. **getList():** In danh sách ra màn hình.



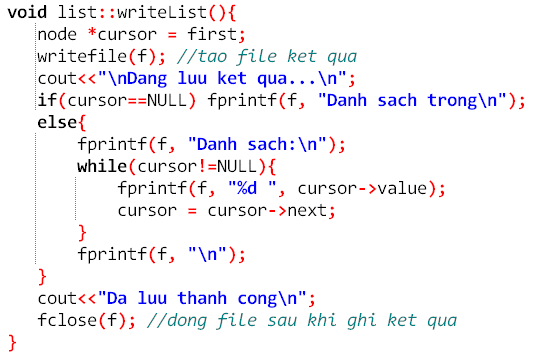
* 1. **getLast():** Tìm nút cuối danh sách.



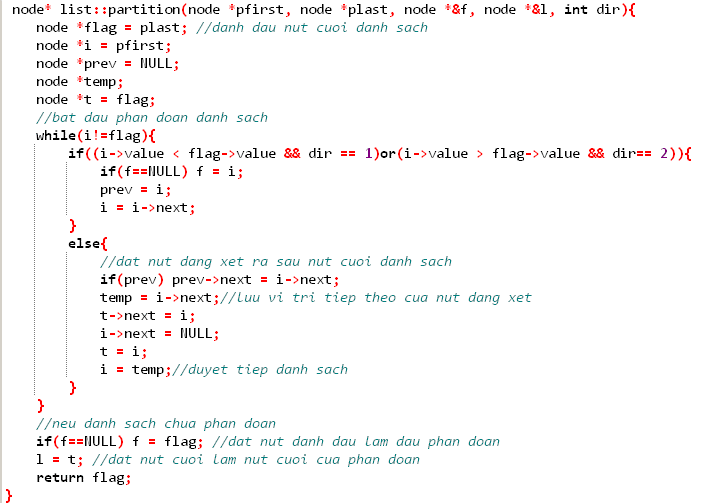
* 1. **swapData():** Đảo giá trị hai nút.



* 1. **writeList():** Ghi danh sách vào file kết quả.

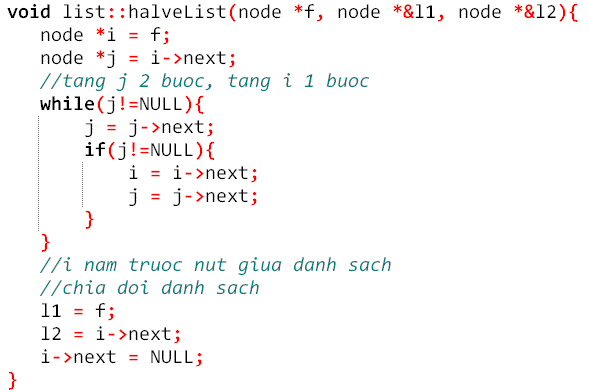


* 1. **partition():** Tạo phân đoạn cho sắp xếp nhanh.
     + Chọn nút cuối đoạn danh sách làm khóa.
     + Duyệt từ đầu danh sách, tìm các nút lớn/nhỏ hơn khóa.
     + Trong quá trình duyệt, di chuyển các nút tìm được ra sau đoạn danh sách.

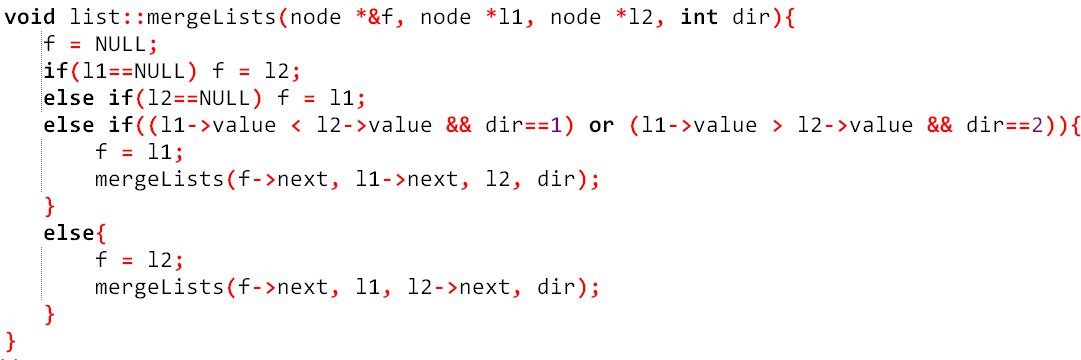


2. 10. **halveList():** Chia đôi đoạn danh sách.

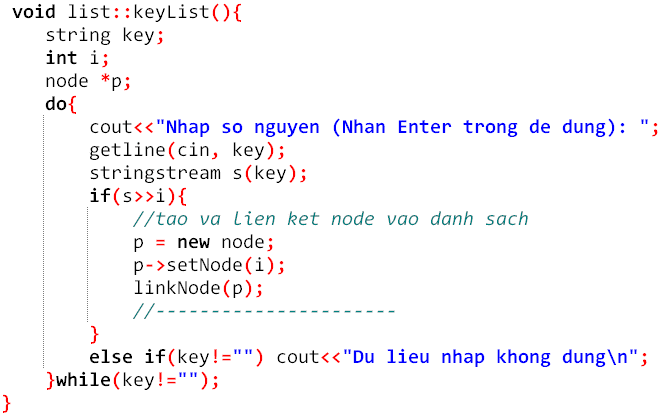
* Duyệt đoạn danh sách có n nút với hai con trỏ i, j. Khi i tăng 1 bước thì j tăng 2 bước.
* Khi j nằm cuối danh sách thì i ở nút thứ n/2.
* Tạo hai đoạn danh sách mới: một có đầu là nút đầu của đoạn danh sách cũ, một có đầu là nút thứ i.
* Đặt nút thứ i+1 là NULL để cắt liên kết giữa hai đoạn danh sách.



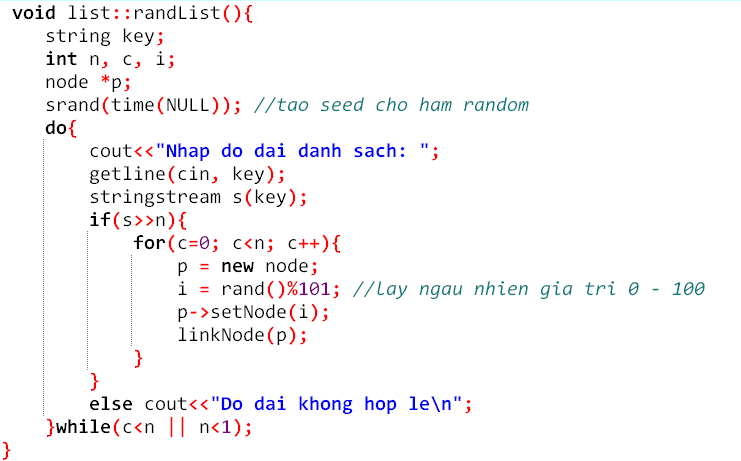
* 1. **mergeLists():** Trộn hai đoạn danh sách.
* Tại một vị trí, xét hai nút từ hai đoạn danh sách khác.
* Nếu một trong hai nút là rỗng thì đặt nút còn lại vào vị trí.
* Nếu hai nút đều có giá trị, so sánh hai nút và đặt nút có giá trị nhỏ hơn vào vị trí hiện tại (nếu sắp xếp tăng dần) rồi lặp lại quá trình trộn cho vị trí tiếp theo (nút chưa được đặt vào vị trí nào sẽ tiếp tục được xét trong lần trộn tiếp theo).
* Quá trình trộn kết thúc khi một trong hai nút rỗng vì hai đoạn danh sách được hiểu là có độ dài gần bằng nhau (chênh lệch không quá một nút).



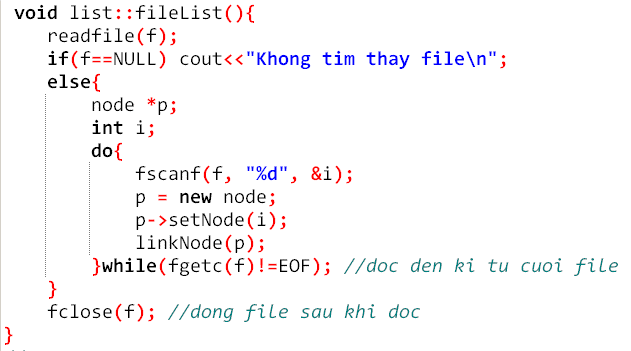
1. **Các phương thức nhập, xuất dữ liệu:**
   1. **keyList():** Nhập dữ liệu từ bàn phím.



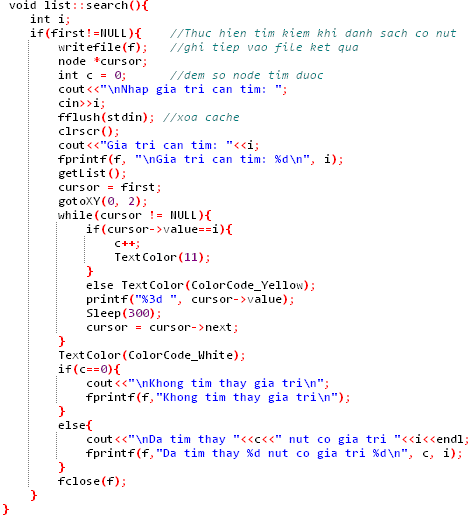
* 1. **randList():** Sinh dữ liệu ngẫu nhiên.



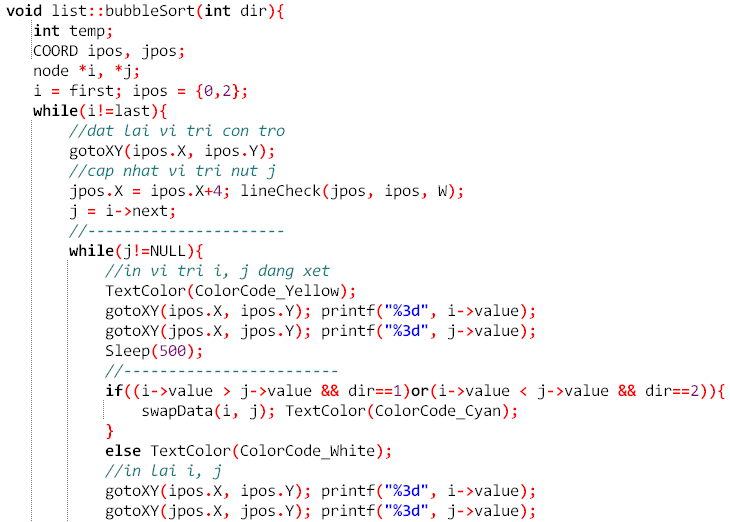
* 1. **fileList():** Nhập dữ liệu từ file.

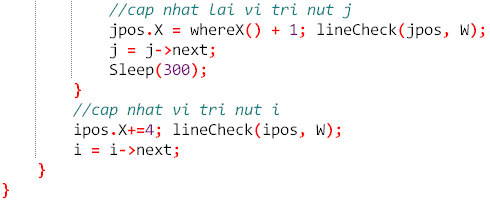


1. **Phương thức tìm kiếm:**

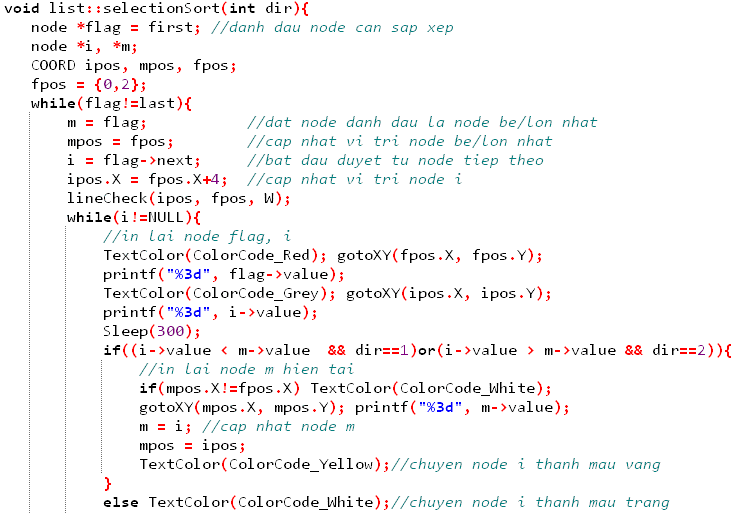


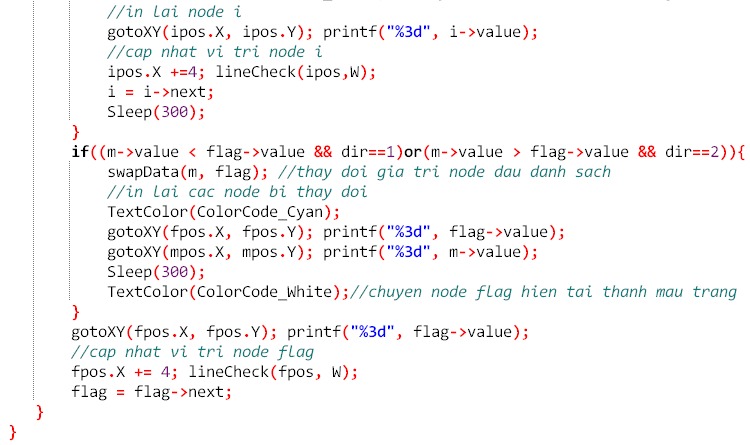
1. **Các phương thức sắp xếp:**
   1. **bubbleSort():**
      * Ở nút i: so sánh nút thứ i với các nút từ sau nó đến cuối danh sách.
      * Đảo vị trí hai nút theo yêu cầu sắp xếp.
      * Quá trình lặp từ đầu đến nút đứng trước nút cuối danh sách;



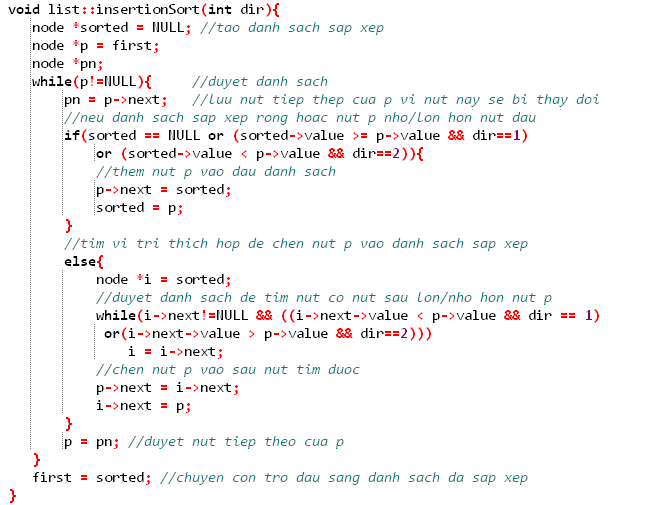


* 1. **selectionSort():**
     + Ở nút i: tìm nút lớn/nhỏ nhất trong đoạn danh sách còn lại.
     + Đảo vị trí nếu nút tìm được khác nút i.
     + Quá trình lặp lại từ đầu đến trước nút cuối danh sách.

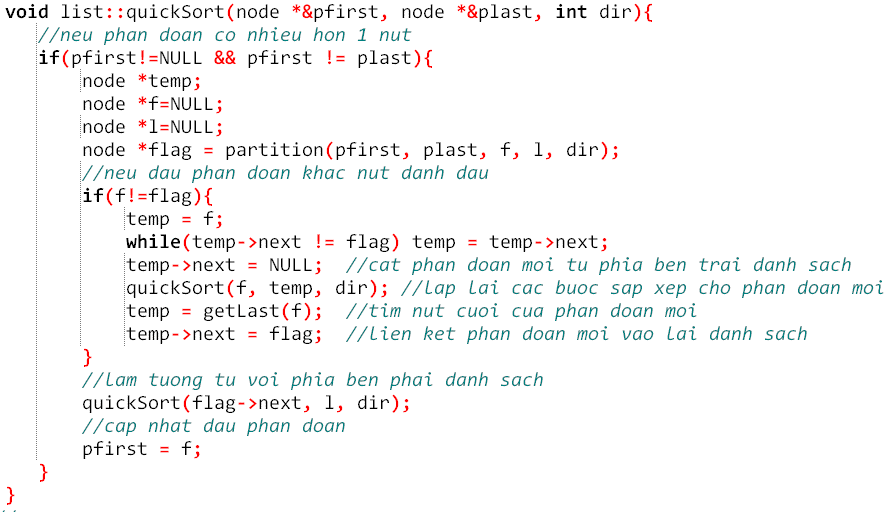




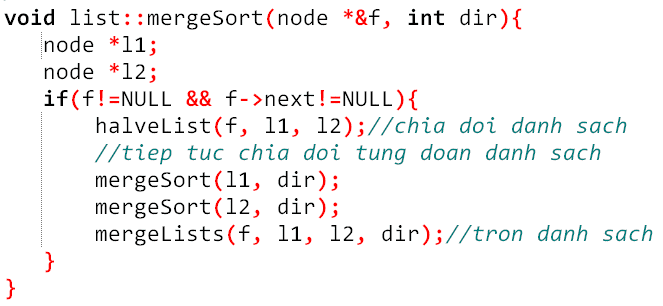
* 1. **insertionSort():**
     + Tạo danh sách rỗng rồi duyệt danh sách cũ.
     + Ở nút thứ p: nếu danh sách mới chưa có nút thì đặt p làm đầu danh sách. Ngược lại thì so sánh nút p với các nút i trong danh sách mới: tìm nút i+1 có giá trị nhỏ hơn p (nếu sắp xếp tăng dần) rồi chèn nút p vào sau nút i+1.
     + Quá trình lặp lại đến khi danh sách cũ rỗng, đặt con trỏ đầu của danh sách cũ sang danh sách mới.



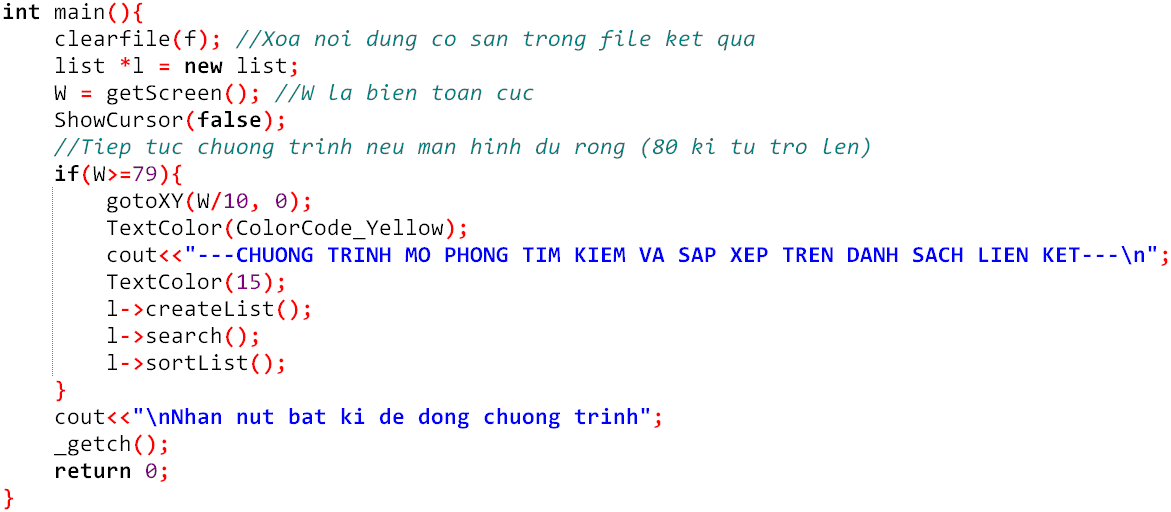
* 1. **quickSort():**
* Trên một đoạn danh sách: nếu có nhiều hơn một nút thì tiến hành phân đoạn.
* Sau khi phân đoạn, cắt phân đoạn bên trái, lặp lại quá trình sắp xếp với phân đoạn này rồi liên kết phân đoạn vào lại danh sách.
* Lặp lại quá trình sắp xếp với phân đoạn bên phải rồi xác định lại nút đầu của đoạn danh sách.



* 1. **mergeSort():**
     + Trên một đoạn danh sách, chia đôi đoạn thành hai danh sách l1, l2.
     + Lặp lại quá trình sắp xếp trên hai nửa đoạn danh sách để tiếp tục chia đôi đến khi mỗi nửa đoạn còn một nút.
     + Tiến hành trộn các nửa đoạn danh sách đã chia.



1. **Chương trình chính:**

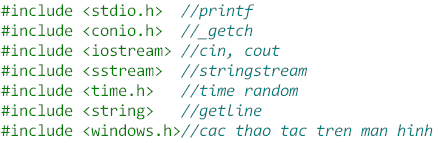


1. **Thư viện sử dụng:**
   1. **Nguồn:**

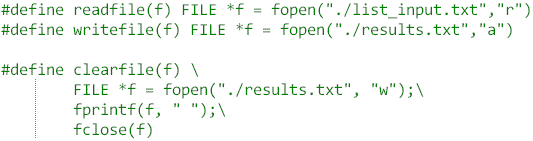
Nguyễn Trung Thành – Thư viện console .h

http://www.mediafire.com/file/kt5r7zwl14swhn6/console.zip/file

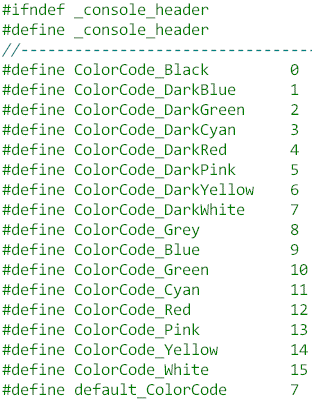
* 1. **Các thư viện dựng sẵn:**

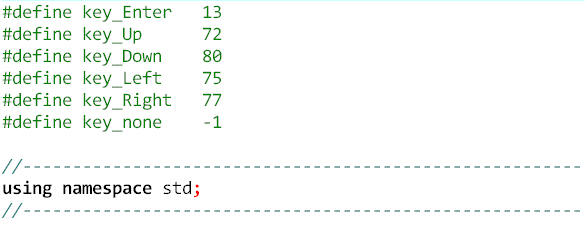


* 1. **Các macro:**

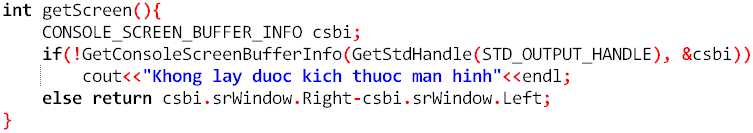


* 1. **Các hằng số :**

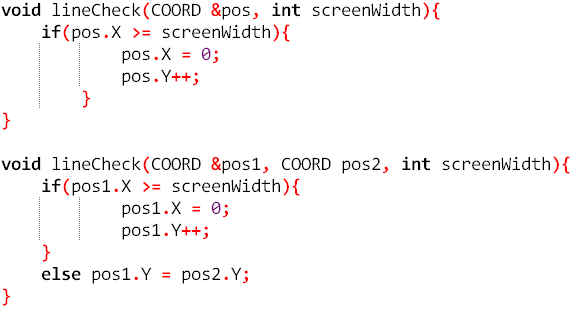




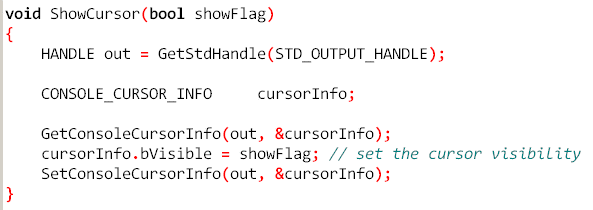
* 1. **Các hàm được sử dụng trong thư viện:**
     + **clrscr():** Xóa màn hình.
     + **gotoXY():** Di chuyển con trỏ màn hình đến tọa độ (X,Y).
     + **whereX():** Hoành độ của con trỏ màn hình.
     + **whereY():** Tung độ của con trỏ màn hình.
     + **TextColor():** Đổi màu chữ.
     + **getScreen():** Độ rộng màn hình.



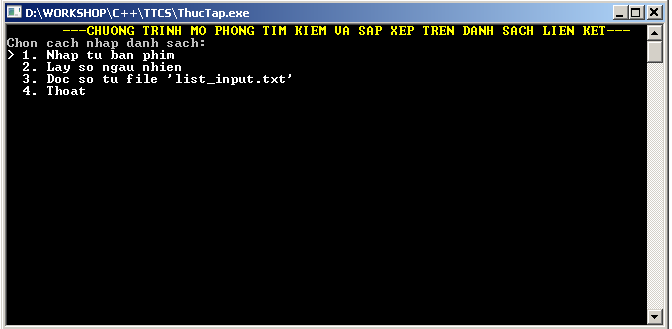
* + - **lineCheck():** Kiểm tra xuống dòng.



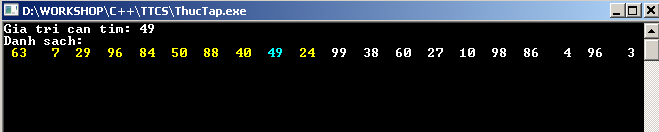
* **ShowCursor():** Bật, tắt con trỏ màn hình.



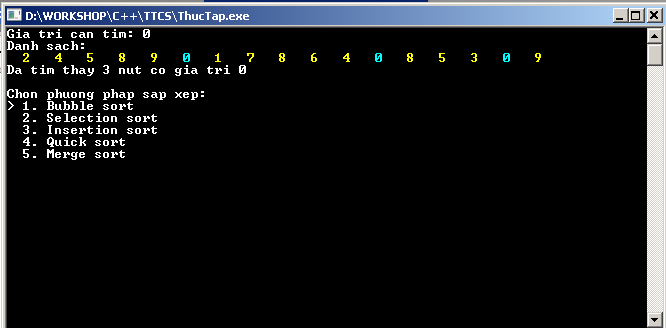
1. **KẾT QUẢ THỰC HIỆN**
   1. **Kết quả đạt được:**
      * Thiết kế lớp.
      * Cài đặt các phương thức nhập, xuất dữ liệu.
      * Cài đặt phương thức tìm kiếm.
      * Cài đặt các phương thức sắp xếp:
        + Bubble sort.
        + Selection sort.
        + Insertion sort.
        + Quick sort.
        + Straight merge sort.
      * Minh họa:
        + Các menu chọn.
        + Phương thức tìm kiếm.
        + Phương thức Bubble sort.
        + Phương thức Selection sort.
   2. **Một số hình ảnh:**
      * Màn hình chính:



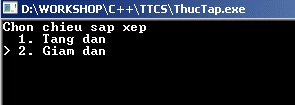
* + - Màn hình tìm kiếm:



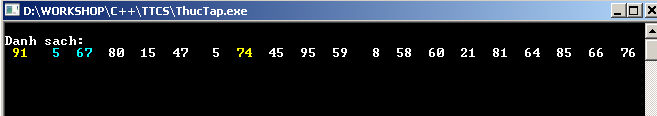
* + - Màn hình lựa chọn các phương pháp sắp xếp:



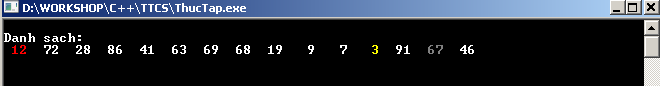
* + - Màn hình lựa chọn chiều sắp xếp:



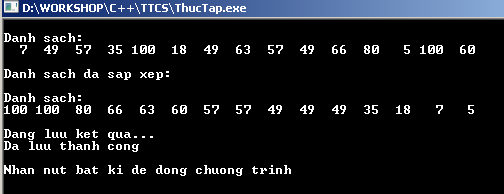
* + - Màn hình sắp xếp bubble sort:



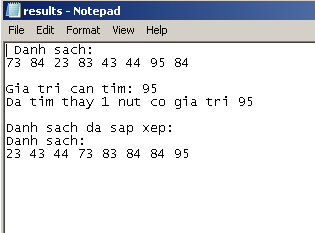
* + - Màn hình sắp xếp selection sort:



* + - Màn hình kết thúc sau khi sắp xếp:



* File kết quả sau khi sắp xếp:



1. **TÀI LIỆU THAM KHẢO:**
2. [**RachitKumarTewari**](https://auth.geeksforgeeks.org/user/RachitKumarTewari/articles) – **Selection sort**   
   <https://www.geeksforgeeks.org/iterative-selection-sort-for-linked-list>.
3. [**Balasubramanian.N**](http://in.linkedin.com/pub/balasubramanian-nagasundaram/3a/361/97b)– **Quick Sort**   
   [https://www.geeksforgeeks.org/quicksort-on-singly-linked-list](https://www.geeksforgeeks.org/quicksort-on-singly-linked-list/).
4. **Quick Sort** – https://www.tutorialcup.com/linked-list/quick-sort-sIngly-linked-list.htm.
5. **Tham biến và con trỏ**   
   https://stackoverflow.com/questions/3834067/c-difference-between-and-in-parameter-passing.
6. **Merge sort**   
   [https://www.geeksforgeeks.org/merge-sort-for-linked-list](https://www.geeksforgeeks.org/merge-sort-for-linked-list/) , <https://stackoverflow.com/questions/14805936/optimal-quicksort-for-single-linked-list>.
7. **Insertion sort** – https://www.geeksforgeeks.org/insertion-sort-for-singly-linked-list.
8. **Kiểm tra số nhập từ bàn phím**   
   <https://stackoverflow.com/questions/25522890/keep-reading-numbers-until-an-empty-input> .
9. **Nguyễn Đức Thuần** – **Bài giảng cấu trúc dữ liệu và giải thuật** – Đại học Nha Trang.
10. **Nguyễn Thị Kim Ngoan** – **Bài giảng lập trình hướng đối tượng** – Đại học Nha Trang.
11. [**Bùi Đức Dương**](http://thuvien.ntu.edu.vn/Opac/BrowseByHeading.aspx?mnuid=141&search_field=AUTHOR&search_id=2569) – **Bài giảng Kỹ thuật lập trình** – Đại học Nha Trang.
12. **Lấy thông số kích thước màn hình console C++** <https://stackoverflow.com/questions/8627327/how-to-get-number-of-characters-in-line-in-console-that-my-process-is-bind-to>.
13. **Bật tắt con trỏ màn hình**

https://stackoverflow.com/questions/18028808/remove-blinking-underscore-on-console-cmd-prompt.