**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH NÂNG CAO**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÍ GIỎ HÀNG**

**Giảng viên:** TRẦN THỊ DUNG

**Sinh viên thực hiện:**

NGUYỄN THỊ NGỌC HIỀN – MSSV: 6051071040

NGUYỄN THỊ DIỄM TRINH – MSSV: 6051071129

TRẦN THỊ THU TRANG – MSSV: 6051071125

**Lớp:**  CÔNG NGHỆ THÔNG TIN K60

**Khoá:** K60

Tp.Hồ Chí Minh, ngày 23 tháng 6 năm 2020

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH NÂNG CAO**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÍ GIỎ HÀNG**

**Giảng viên:** TRẦN THỊ DUNG

**Sinh viên thực hiện:**

NGUYỄN THỊ NGỌC HIỀN – MSSV: 6051071040

NGUYỄN THỊ DIỄM TRINH – MSSV: 6051071129

TRẦN THỊ THU TRANG – MSSV: 6051071125

**Lớp:** CÔNG NGHỆ THÔNG TIN K60

**Khoá:** K60

Tp. Hồ Chí Minh, ngày 23 tháng 6 năm 2020

|  |
| --- |
|  |

**MỤC LỤC**

Nội dung

[LỜI CẢM ƠN 1](#_Toc43956532)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc43956533)

[**1.** **ĐẶT VẤN ĐỀ** 2](#_Toc43956534)

[**1.1.** **Lý do chọn đề tài** 2](#_Toc43956535)

[**1.2.** **Mục tiêu nghiên cứu** 2](#_Toc43956536)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc43956537)

[**I.** **Các thuật toán hỗ trợ** 3](#_Toc43956538)

[**1.** **Đọc – Ghi file** 3](#_Toc43956539)

[**2.** **Cấu trúc liên kết đơn** 6](#_Toc43956540)

[**3.** **Thuật toán sắp xếp** 10](#_Toc43956541)

[**4.** **Thuật toán tìm kiếm** 13](#_Toc43956542)

[**I.** **Xác định yêu cầu** 15](#_Toc43956543)

[**1.** **Yêu cầu người dùng** 15](#_Toc43956544)

[**2.** **Yêu cầu phần mềm** 15](#_Toc43956545)

[**II.** **Phân tích chương trình** 15](#_Toc43956546)

[**1.** **Giải thích code** 15](#_Toc43956547)

[**III.** **Xây dựng chương trình quản lý giỏ hàng** 19](#_Toc43956548)

[**1.** **Hướng dẫn sử dụng** 19](#_Toc43956549)

[**2.** **Các form trong chương trình** 20](#_Toc43956550)

[TỔNG KẾT 21](#_Toc43956551)

[**Kết quả đạt được** 21](#_Toc43956552)

[TÀI LIỆU THAM KHẢO 22](#_Toc43956553)

[PHỤ LỤC 23](#_Toc43956554)

[**Phụ lục 1: Hướng dẫn cài đặt** 23](#_Toc43956555)

# **LỜI CẢM ƠN**

Lời đầu tiên cho nhóm chúng em gửi đến thầy cô đã giảng dạy cho chúng em trong thời gian qua. Những người đã truyền đạt cho chúng em rất nhiều kiến thức hay và có ích để chúng em có thể hoàn thiện bản thân mình hơn cũng như có thêm nhiều kiến thức hơn về cuộc sống. Tiếp đến cho chúng em gửi lời cảm ơn đến thầy cố vấn học tập Nguyễn Lê Minh đã giảng dạy môn Tin học đại cương, thầy đã tạo cho chúng em một nền tảng tốt để khi chúng em tiếp cận với môn lập trình nâng cao được dễ dàng hơn. Và chúng em cũng cảm ơn cô Trần Thị Dung, giảng viên giảng dạy môn lập trình nâng cao đã giúp chúng em xây dựng được một hệ thống kiến thức vững chắc để chúng em có thể hoàn thành một cách tốt nhất bài báo cáo lần này. Trước những điều kiện không mấy thuận lợi trong quá trình học do dịch bệnh Covid-19 diễn biến khá phức tạp, với việc học online làm cho chúng em khá khó khăn trong việc tiếp cận với kiến thức, nhưng với sự giảng dạy tận tình và đầy nhiệt huyết của cô làm cho chúng em có thể dễ dàng tiếp thu những kiến thức nền tảng bổ ích. Với thời gian ngắn ngủi cũng như những kiến thức còn nhiều thiếu sót của chúng em, em mong cô có thể bỏ qua và tạo điều kiện tốt nhất cho nhóm chúng em để chúng em hoàn thành tốt bài báo cáo này.

# **CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI**

1. **ĐẶT VẤN ĐỀ**
   1. **Lý do chọn đề tài**

Trong xã hội ngày càng phát triển hiện nay, khoa học công nghệ là thứ không thể thiếu với mỗi quốc gia, doanh nghiệp, trường học hay mỗi cá nhân đặc biệt là công nghệ thông tin. Với sự phát triển nhanh chóng không ngừng nghỉ như vậy của công nghệ thông tin đã giúp giải quyết các công việc học tập, nghiên cứu, quản lý thông tin… Một cách dễ dàng và tiện lợi. Thấy được tiềm năng đó các quốc gia, doang nghiệp, trường học, cá nhân… đã ứng dụng nó vào thực tiễn cuộc sống để giải quyết công việc, học tập giải trí chỉ với một chiếc điện thoại thông minh.

Trong những năm gần đây, nhu cầu về các phần mềm ứng dụng học tập, giải trí ngày càng nhiều hơn, do nhu cầu sử dụng điện thoại thông minh, các thiết bị gọn nhẹ ngày càng cao và tiện dụng 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 người già.

Nhờ vào sự phát triển của những ứng dụng công nghệ, ngày càng nhiều ứng dụng được xuất hiện, trong đó có các ứng dụng mua sắm trực tuyến như shopee, tiki, lazada… và nhu cầu mua sắm trực tuyến của con người cũng ngày càng tăng cao hơn. Mọi người chỉ cần ở nhà hoặc bất kì nơi đâu, chỉ cần những chiếc điện thoại thông minh của họ có kết nối internet, không cần đến siêu thị hay các cửa hàng mà người tiêu dùng đã có thể mua sắm, đi chợ trên những ứng dụng này. Khi đi chợ trực tuyến như vậy thì giỏ hàng là một phần không thể thiếu để người tiêu dùng có thể thêm hoặc bỏ bớt đi một mặt hàng nào đó trong đơn hàng của mình. Do đó phầm mềm quản lý giỏ hàng là một phần không thể thiếu trong các ứng dụng mua sắm trực tuyến để cho người tiêu dùng và người bán hàng có thể kiểm soát tốt được những mặt hàng gì mà họ đang mua và bán.

* 1. **Mục tiêu nghiên cứu**
* Xây dựng phần mềm quản lý giỏ hàng bằng ngôn ngữ C.
* ứng dụng phải đáp ứng được giao diện thân thiện với người dùng, đầy đủ các chức năng thêm, hủy đơn hàng.
* Hiểu rõ hơn về giỏ hàng của mình khi mua sắm trực tuyến.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

1. **Các thuật toán hỗ trợ**
2. **Đọc – Ghi file**
   1. **Các kiểu file**
      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, chúng 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.

* + 1. **File nhị phân (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. Bạn 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. **Các thao tác với file**
     1. **Mở file trong C**

Sử dụng hàm **fopen()** để tạo file mới hoặc để mở các file đã tồn tại. Cách gọi này sẽ khởi tạo đối tượng loại FILE, mà bao gồm thông tin cần thiết để điều khiển luồng. Dưới đây là một cách gọi hàm:

FILE \*fopen(const char \* ten\_file, const char \* mode);

Trong đó, **ten\_file** là một chuỗi, được coi như tên file và giá trị **mode** truy cập có thể là những giá trị dưới đây:

|  |  |
| --- | --- |
| **Mode** | **Miêu tả** |
| r | Mở các file đã tồn tại với mục đích đọc |
| w | Mở các file với mục đích ghi. Nếu các file này chưa tồn tại thi file mới được tạo ra. Ở đây, chương trình bắt đầu ghi nội dung từ phần mở đầu của file |
| a | Mở file văn bản cho việc ghi ở chế độ ghi tiếp theo vào cuối, nếu nó chưa tồn tại thì file mới được tạo. Đây là chương trình ghi nội dung với phần cuối của file đã tồn tại. |
| r+ | Mở file văn bản với mục đích đọc và ghi. |
| w+ | Mở một file văn bản cho chế độ đọc và ghi. Nó làm trắng file đã tồn tại nếu file này có và tạo mới nếu file này chưa có. |
| a+ | Mở file đã tồn tại với mục đích đọc và ghi. Nó tạo file mới nếu không tồn tại. Việc đọc file sẽ bắt đầu đọc từ đầu nhưng ghi file sẽ chỉ ghi vào cuối file. |

* + 1. **Đóng file trong C**

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

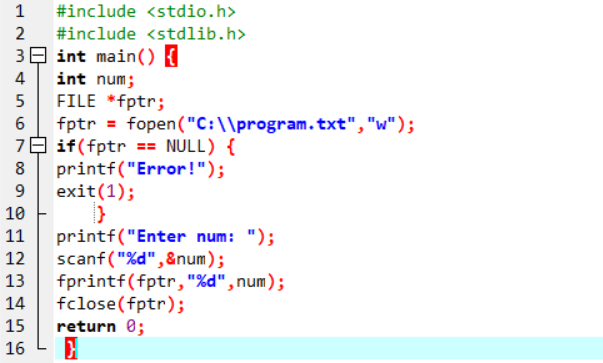
Việc đóng file đang mở có thể được thực hiện bằng cách dùng hàm fclose().

fclose(fptr);

* + 1. **Đọc/Ghi file** 
       1. **Đọc/Ghi file văn bản trong C**

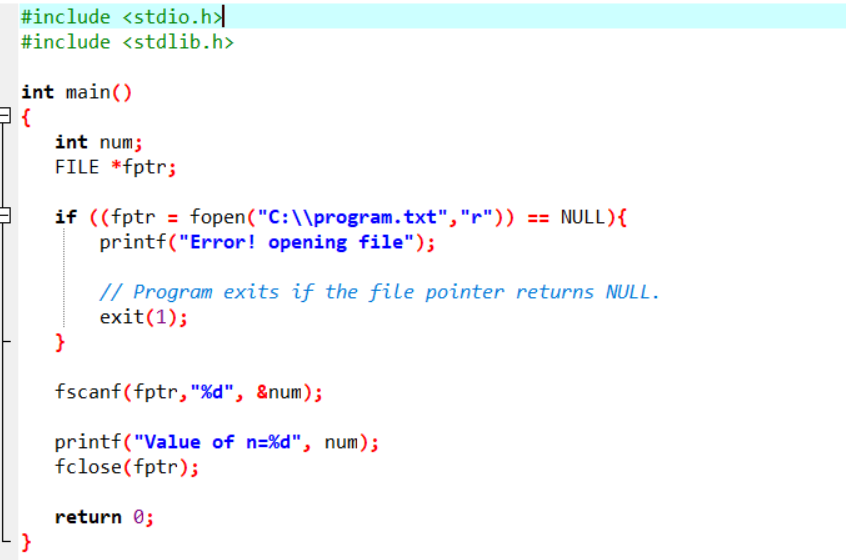
Đầu tiên là đọc ghi file trong C với file văn bản. Để làm việc với file văn bản, chúng ta sẽ sử dụng **fprintf()** và **fscanf().**

VD**:** Ghi file sử dụng **fprintf()**



**Hình 1.**

VD: Đọc file sử dụng **fscanf()**



**Hình 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.

* + - 1. **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.

1. *Ghi file nhị phân*

Để ghi file nhị phân, chúng ta 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 chúng ta muốn ghi.



1. *Đọc file nhị phân*

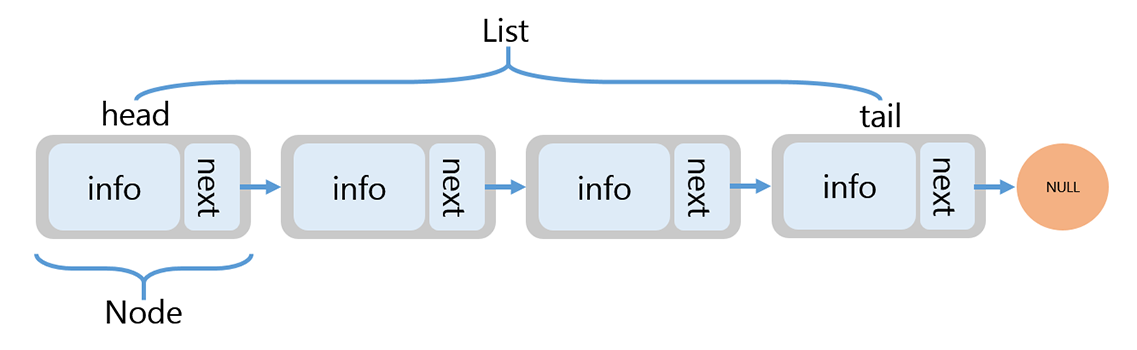
Hàm **fread()** cũng có 4 tham số tương tự như hàm **fwrite()** phía trên.



1. **Cấu trúc liên kết đơn**
   1. **Danh sách liên kết là gì?**

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.

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

****

**Hình 3.**

Danh sách các kiểu danh sách liên kết:

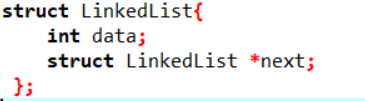
• Danh sách liên kết đơn(Single linked list): Chỉ có sự kết nối từ phần tử phía trước tới phần tử phía sau.

• 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.

* 1. **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 4.**

Khai báo trên sẽ được sử dụng cho mọi Node trong linked list. Trường data sẽ lưu giữ 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.

* + 1. **Thêm vào đầu**

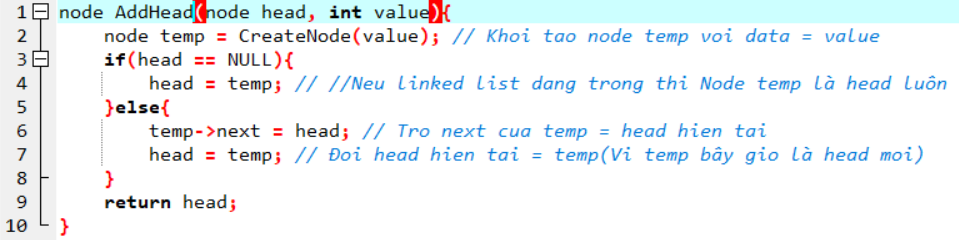
Việc thêm vào đầu chính là việc cập nhật lại head. Ta gọi Node mới(temp), ta có:

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



**Hình 5.**

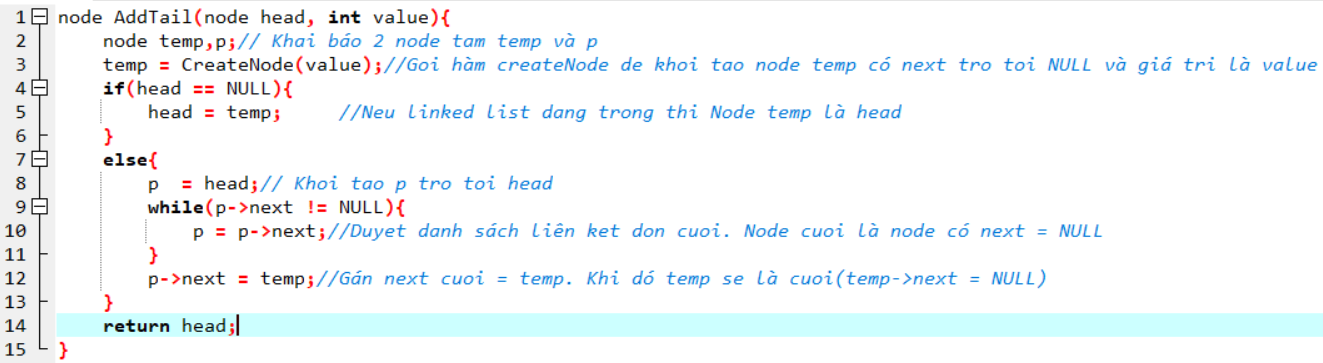
* + 1. **Thêm vào cuối**

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

Tạo một Node mới với giá trị value

Nếu head = NULL, tức là danh sách liên kết đang trống. Khi đó Node mới (temp) sẽ là head luôn.

Ngược lại, ta sẽ duyệt tới Node cuối cùng (Node có next = NULL), và trỏ next của cuối tới Node mới (temp).



**Hình 6**.

* + 1. **Xóa đầu**

Xóa 1 phần tử ở đầu danh sách không chỉ đơn giản là cập nhập lại biến con trỏ pHead, mà ta phải giải phóng được vùng nhớ của Node cần xóa.

**Bước 1:** Khai báo một con trỏ p để lưu lại địa chỉ của Node đầu tiên.

SNode\* p = pHead;

**Bước 2:** cập nhập lại giá trị của pHead.

pHead = pHead->pNext;

**Bước 3:** giải phóng vùng nhớ của Node cần xóa.

delete p;

A screenshot of a cell phone

Description automatically generated

**Hình 7.**

* + 1. **Xóa cuối**

Tương tự như thao tác xóa phần tử đầu danh sách, để xóa một phần tử đứng sau phần tử q, ta thực hiện các bước sau:

Bước 1: Khai báo một con trỏ p để lưu lại địa

chỉ của Node đầu tiên.

SNode\* p = pTail;

Bước 2: cập nhập lại giá trị của pTail.

pTail = pTail->pNext;

Bước 3: giải phóng vùng nhớ của Node cần xóa.

delete p;

A screenshot of a cell phone

Description automatically generated

**Hình 8.**

1. **Thuật toán sắp xếp**
   1. **Thuật toán sắp xếp nổi bọt (Bubble Sort)**
      1. Ý tưởng thuật toán

Sắp xếp dãy số bằng cách lặp lại công việc đổi chỗ hai số đứng liên tiếp nhau nếu chúng đứng sai thứ tự (vd: đổi chỗ hai số nếu số sau bé hơn số trước trong trường hợp sắp xếp tăng dần).

VD: Giả sử sắp xếp dãy số [8 4 3 6 7] tăng dần

* + Lần lặp đầu tiên:

[**8 4** 3 6 7] 🡪 [**4 8** 3 6 7]: ở đây sẽ so sánh hai số đầu tiên trong dãy số, và đổi chỗ cho nhau nếu số sau bé hơn số trước (**8** > **4**)

[4 **8 3** 6 7] 🡪 [4 **3 8** 6 7]: Đổi chỗ 8->3

[4 3 **8 6** 7] 🡪 [4 3 **6 8** 7]: Đổi chỗ 8->6

[4 3 6 **8 7**] 🡪 [4 3 6 **7 8**]: Đổi chỗ 8->7

* + Lần lặp thứ hai:

[**4 3** 6 7 8] 🡪 [**3 4** 6 7 8]: Đổi chỗ 4->3

[3 **4 6** 7 8] 🡪 [3 **4 6** 7 8]

[3 4 **6 7** 8] 🡪 [3 4 **6 7** 8]

[3 4 6 **7 8**] 🡪 [3 4 6 **7 8**]

* + Lần lặp thứ ba:

Lúc này dãy số đã được sắp xếp nhưng thuật toán không thể nhận ra ngay được mà phải kiểm tra thêm một lần nữa để chắc chắn dãy số đã được sắp xếp, để khi dò các phần tử từ đầu đến cuối không có sự đổi chỗ nào xảy ra nữa.

[**3 4** 6 7 8] 🡪 [**3 4** 6 7 8]

[3 **4 6** 7 8] 🡪 [3 **4 6** 7 8]

[3 4 **6 7** 8] 🡪 [3 4 **6 7** 8]

[3 4 6 **7 8**] 🡪 [3 4 6 **7 8**]

* + 1. Cài đặt thuật toán

VD: Sắp xếp dãy số tăng dần, giảm dần

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

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

for(int j=i+1;j>i;j--){

if(a[j] < a[i]){

HoanVi(a[i],a[j]);

}

}

}

}

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

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

for(int j=i+1;j>i;j--){

if(a[j] > a[i]){

HoanVi(a[i],a[j]);

}

}

}

}

void HoanVi(int &a,int &b){

int temp = a;

a=b;

b=temp;

}

* 1. **Thuật toán sắp xếp chọn (Selection Sort)**
     1. Ý tưởng thuật toán

Sắp xếp dãy số dựa trên việc so sánh tại chỗ, trong đó danh sách được chia thành hai phần, phần đã sắp xếp là danh sách trống ở bên trái và phần chưa sắp xếp là toàn bộ danh sách ban đầu.

Chọn trong phần chưa sắp xếp phần tử nhỏ nhất và di chuyển phần tử đó về vị trí đầu của phần bên trái, sau đó xét tiếp các phần tử còn lại của phần bên phải. Tiến trình này cứ tiếp tục cho đến khi toàn bộ phần tử của phần chưa sắp xếp được di chuyển về phần đã sắp xếp.

* + 1. Cài đặt thuật toán

void sapxep(int a[], int n)

{

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

int min = i;

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

if (a[min]>a[j]){

min = j;

HoanVi(a[i], a[min]);

}

}

}

}

void HoanVi(int &a,int &b){

int temp = a;

a=b;

b=temp;

}

* 1. **Thuật toán sắp xếp chèn (Insertion Sort)**
     1. Ý tưởng thuật toán

Sắp xếp dãy số dựa trên việc so sánh tại chỗ, duyệt từng phần tử và chèn phần tử đó vào mảng con đã sắp xếp (là mảng gồm phần tử đầu đến phần tử trước nó) sao cho phần tử đó lớn hơn phần tử trước nó và nhỏ hơn phần tử kế sau (với trường hợp tăng dần) hoặc ngược lại (với trường hợp giảm dần). Tiến trình này tiếp tục cho đến khi tất cả phần tử trong mảng được chèn đến đúng vị trí của nó và vẫn đảm bảo được tính chất của một dãy tăng dần hoặc giảm dần.

* + 1. Cài đặt thuật toán

void insertionSort(int a[], int n)

{

int i, k, j;

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

{

k = a[i];

j = i-1;

/\* Di chuyển các phần tử có giá trị lớn hơn giá trị k về sau một vị trí so với vị trí ban đầu của nó \*/

while (j >= 0 && a[j] > k)

{

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

j = j-1;

}

a[j+1] = k;

}

}

1. **Thuật toán tìm kiếm**
   1. **Thuật toán tìm kiếm tuyến tính (Linear Search)**
      1. Ý tưởng thuật toán

Tìm kiếm một phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần từ của danh sách đó đến khi nào tìm được giá trị mong muốn hay đã duyệt hết qua hết danh sách.

* + 1. Cài đặt thuật toán

int TimKiemTuyenTinh(int a[], int n, int x)

{

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

if (a[i] == x){

return 1;

}

}

return 0;

}

* 1. **Thuật toán tìm kiếm nhị phân (Binary Search)**
     1. Ý tưởng thuật toán
     + Trường hợp sử dụng thuật toán này với mảng đã sắp xếp (tăng hoặc giảm dần)

B1: Xét đoạn mảng arr[left…right] cần tìm kiếm phần tử x. Ta so sánh x với phần tử ở vị trí giữa của mảng(mid = (left + right)/2). Nếu:

B2: Nếu phần tử arr[mid] = x. Kết luận và thoát chương trình.

B3: Nếu arr[mid] < x. Chỉ thực hiện tìm kiếm trên đoạn arr[mid+1…right].

B4: Nếu arr[mid] > x. Chỉ thực hiện tìm kiếm trên đoạn arr[left…mid-1].

* + 1. Cài đặt thuật toán

void binarySearch(int a[], int left, int right, int x){

int mid=(left + right)/2;

if(a[mid]==x){

printf("x co trong day!");

}

else if(a[mid]<x){

binarySearch(a,mid+1,right,x);

}

else if(a[mid]>x){

binarySearch(a,left,mid-1,x);

}

else{

printf("x khong co trong day!");

}

}

**CHƯƠNG 3. PHÂN TÍCH VÀ XÂY DỰNG ĐỀ TÀI**

1. **Xác định yêu cầu**
2. **Yêu cầu người dùng**
   * Yêu cầu về nhập xuất dữ liệu: người dùng có thể nhập dữ liệu từ bàn phím và xuất dữ liệu ra màn hình console
3. **Yêu cầu phần mềm**
   * Chương trình chạy được
   * Viết bằng ngôn ngữ C, sử dụng phương pháp hàm
   * Chương trình phải đầy đủ các chức năng như: nhập, xuất, lưu vào file,..vv..
4. **Phân tích chương trình**
5. **Giải thích code**

Khi chạy, chương trình sẽ lần lượt thực hiện các công việc trong hàm main:

* Trong hàm menu:

- Màn hình console sẽ lần lượt xuất ra các lựa chọn cho người dùng.

- Ở lệnh while:

**while** ( key < 0 || key > 6 ) {

printf("Nhap sai! ");

printf("Vui long nhap 1 so: ");

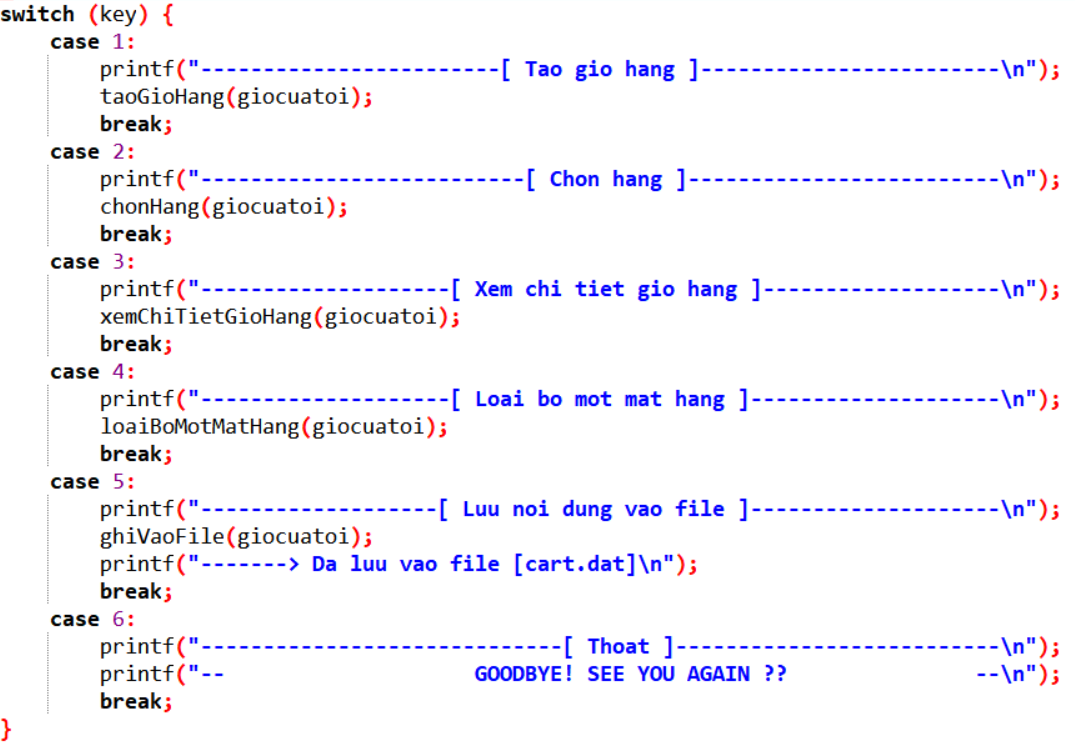
scanf("%d", &key);

}

nếu người dùng nhập từ bàn phím các lựa chọn không có trong hàm menu thì màn hình console sẽ hiện ra “Nhap sai!” và yêu cầu người dùng nhập lại.

- Ở lệnh switch case:

Khi người dùng nhập đúng các lựa chọn ở hàm menu, chương trình sẽ thực hiện các công việc ở các case tương ứng.



**Hình 9.**

Tiếp theo:

**if** (key > 0 && key < 6) **Menu**(giocuatoi);

Khi hoàn thành xong một công việc trong một case đã chọn trước đó, màn hình console hiện ra yêu cầu người dùng chọn một công việc tiếp theo để thực hiện mà không bị thoát chương trình (trường hợp người dùng nhập case không có trong menu thì sẽ quay lại lệnh while), cứ như vậy cho đến khi người dùng thực hiện xong các công việc cần thiết và chọn thoát chương trình.

* Trong hàm taoGioHang(): [\\case](file:///\\case) 1

**void** taoGioHang(giohang\_st &gct) {

printf("Nhap ten khach hang: ");

fflush(stdin);

gets(gct.tenkh);

printf("Nhap dia chi: ");

fflush(stdin);

gets(gct.diachi);

gct.sosp = 0;

}

- Lệnh printf() sẽ xuất ra màn hình và yêu cầu người dùng nhập tên và địa chỉ khách hàng.

- Vì tên và địa chỉ khách hàng là một chuỗi ký tự nên không thể sử dụng lệnh scanf() để nhập mà phải sử dụng lệnh gets(). Hàm fflush(stdin) đặt trước hàm gets() để xóa bộ nhớ đệm.

* Trong hàm chonHang():

**void** chonHang(giohang\_st &giocuatoi) {

printf("Nhap ten san pham: ");

int x = giocuatoi.sosp;

fflush(stdin);

gets(giocuatoi.dssp[x].tensp);

printf("Nhap so luong: ");

scanf("%d", &giocuatoi.dssp[x].soluong);

printf("Nhap gia: ");

scanf("%d", &giocuatoi.dssp[x].gia);

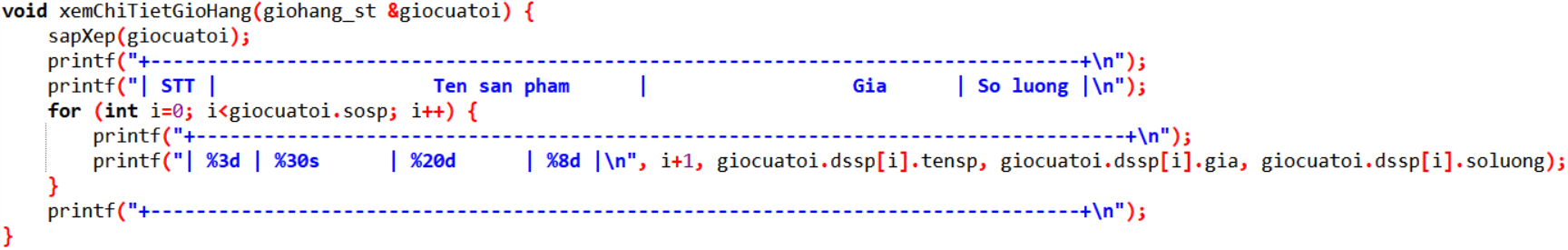
giocuatoi.sosp ++; // sau khi nhap 1 san pham thi so

san pham tang len 1

}

Cũng tương tự như hàm taoGioHang(), hàm chonHang() cũng lần lượt thực hiện những thao tác nhập dữ liệu như trên để hoàn thành công việc của hàm.

* Trong hàm xemChiTietGioHang():



**Hình 10.**

Công việc của hàm này là in ra màn hình console những dữ liệu của sản phẩm mà người dùng đã nhập trước đó ở hàm chonHang().

* Trong hàm sapXep():

Hàm sapXep() được gọi vào trong hàm xemChiTietGioHang(), khi người dùng nhập xong danh sách dữ liệu của sản phẩm sẽ thực hiện sắp xếp danh sách sản phẩm theo tên sản phẩm từ A -> Z.

**void** sapXep(giohang\_st &giocuatoi) {

for (int i=0; i<giocuatoi.sosp; i++)

for (int j=0; j<giocuatoi.sosp; j++)

if (strcmp(giocuatoi.dssp[i].tensp, giocuatoi.dssp[j].tensp) < 0) {

sp\_st tmp = giocuatoi.dssp[i];

giocuatoi.dssp[i] = giocuatoi.dssp[j];

giocuatoi.dssp[j] = tmp;

}

}

* Trong hàm loaiBoMotMatHang():

**void** loaiBoMotMatHang(giohang\_st &giocuatoi) {

printf("Can loai bo san pham thu may: ");

int k;

scanf("%d", &k);

if (k < 1 || k > giocuatoi.sosp)

printf("Ban phai nhap thu tu dung voi danh sach!");

else {

for (int i=k-1; i<giocuatoi.sosp; i++)

giocuatoi.dssp[i] = giocuatoi.dssp[i+1];

giocuatoi.sosp--; // Sau khi xoa 1 san pham thi so san pham - 1;

}

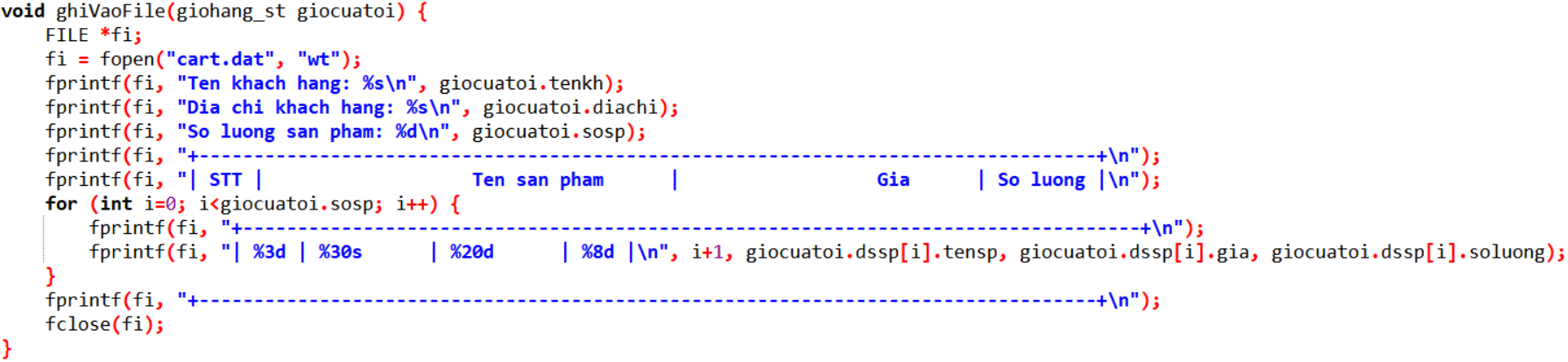
}

Khi chạy chương trình sẽ yêu cầu người dùng người dùng nhập số thứ tự của mặt hàng muốn loại bỏ trong danh sách đã nhập (nhập số k), nếu nhập sai thứ tự của mặt hàng trong danh sách thì chương trình sẽ quay về hàm menu.

* Trong hàm ghiVaoFile():

Lệnh fi = fopen("cart.dat", "wt") dùng để mở file cart.dat và ghi dữ liệu của lần mua hàng vào file, wt có nghĩa là mở file để ghi vào.

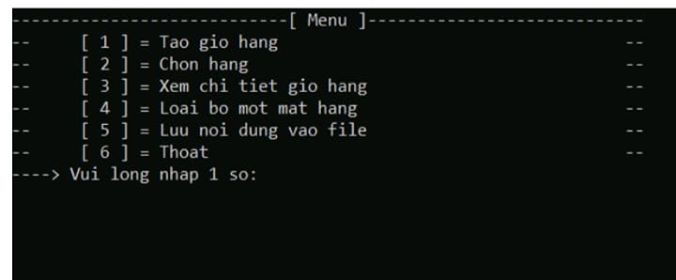
fprintf() dùng để ghi vào file, fclose() dùng để đóng file sau khi ghi xong.



**Hình 11.**

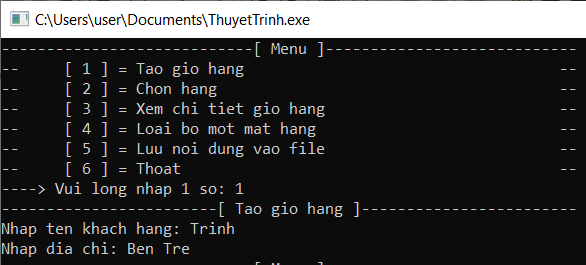
1. **Xây dựng chương trình quản lý giỏ hàng**
2. **Hướng dẫn sử dụng**

Sau khi chạy chương trình, màn hình sẽ hiện lên như sau:

****

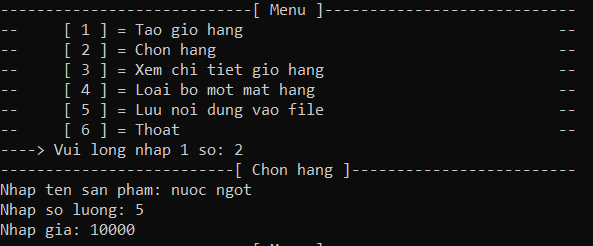
**Hình 12.**

Chọn 1, khách hàng sẽ bắt đầu tạo giỏ hàng. Chương trình sẽ yêu cầu nhập thông tin khách hàng (gồm tên và địa chỉ). Ví dụ:



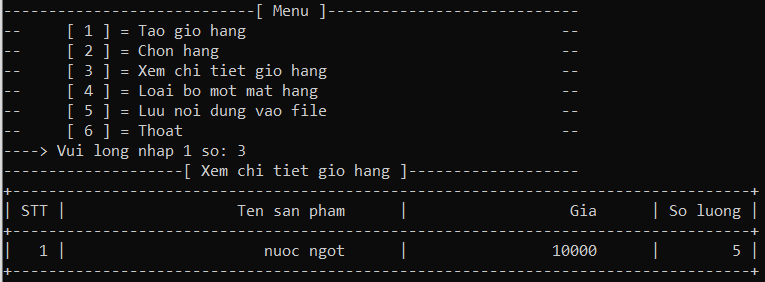
**Hình 13.**

Chọn 2, khách hàng sẽ chọn hàng hóa để mua. Khách hàng cần nhập tên sản phẩm, số lượng và giá của sản phẩm. Ví dụ:



**Hình 14.**

Chọn 3 để xem chi tiết những món hàng trong giỏ hàng.



**Hình 15.**

Để xóa một món hàng trong giỏ hàng, khách hàng cần chọn số 4 và nhập đúng số thứ tự của món hàng.

Chọn 5 để lưu tất cả thông tin vào file và để thoát chương trình, chọn số 6.

1. **Các form trong chương trình**
   * Menu
   * Tạo giỏ hàng
   * Chọn hàng
   * Xem chi tiết giỏ hàng
   * Loại bỏ một mặt hàng
   * Lưu nội dung vào file
   * Thoát chương trình

# **TỔNG KẾT**

## **Kết quả đạt được**

1. **Ưu điểm**

* Đáp ứng được những yêu cầu của phần mềm quản lý giỏ hàng
* Các thành viên thống nhất được với nhau trong việc phân chia công việc

1. **Nhược điểm**

* Chưa tạo giao diện cho chương trình
* Tiến độ hoàn thành công việc của các thành viên nhóm còn chậm

# **TÀI LIỆU THAM KHẢO**

[1]. Lập trình không khó (<https://nguyenvanhieu.vn/>)

[2]. Stdio (<https://www.stdio.vn/>)

[3]. Danh sách liên kết đơn\_ Nguyễn Công Cường([https://cuongblhs.blogspot.com/](https://l.facebook.com/l.php?u=https%3A%2F%2Fcuongblhs.blogspot.com%2F%3Ffbclid%3DIwAR3Na9bGB7iGMRO483zmgBdCUcHNOEpCAknX8WqizgqRRQkRqb9l3H4boqM&h=AT2zqrnZUkpx6KWpmKxEpenzYBFF6Whd4ZlonZZNmlPq6d83NM_sIZlKbundQzUIS-5UQ6jqpRGBH7iWrPlx3B3NTqDup3_w-noy3N_rGgyTAO2H6PzQz2vhNBKPYgY5RqZwoQ))

# **PHỤ LỤC**

## **Phụ lục 1: Hướng dẫn cài đặt**

**Phần mềm chạy chương trình**

Sử dụng 2 phần mềm chính là dev c++ và c free.

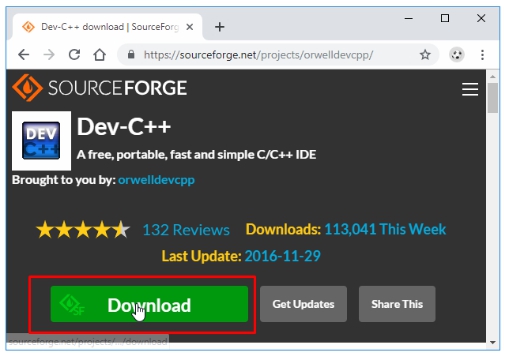
Hướng dẫn tải dev c++:

**Dev C++ là gì?**

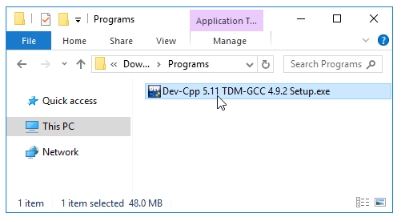
Dev C++ là một IDE, một phần mềm dùng để soạn thảo/ lập trình ngôn ngữ C/C++.

Dev C++ khá nhẹ, tích hợp sẵn các tính năng như highlight (giúp phân biệt hàm, biến…), auto complete (gợi ý code), tự động lưu code, compile và run trực tiếp…

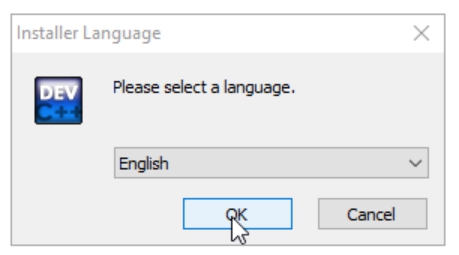
**Cài đặt Dev C++**

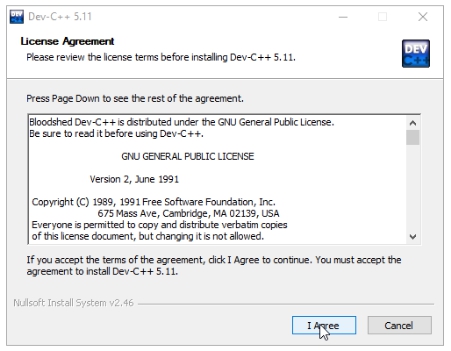


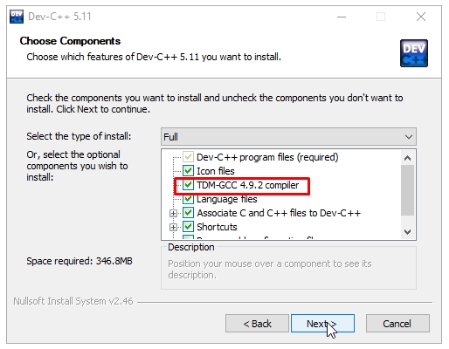
Click đúp vào file .exe vừa tải về

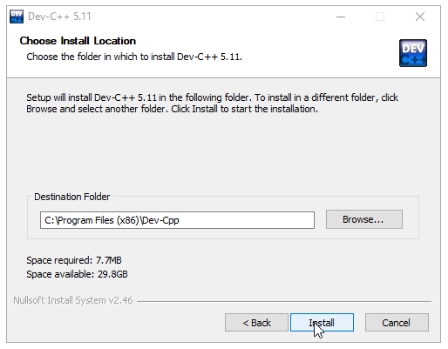


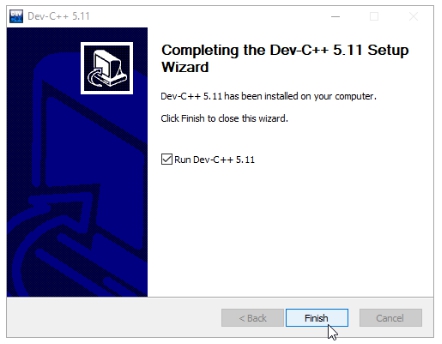
Chọn ngôn ngữ cài đặt



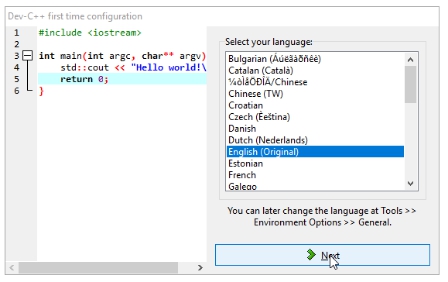


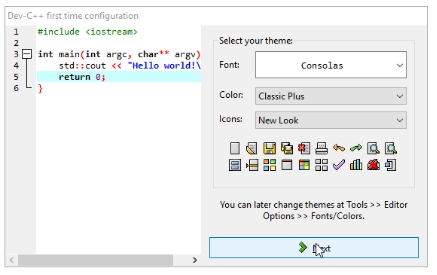


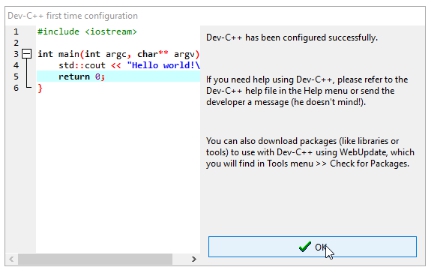




Lần đầu chạy Dev C++ bạn sẽ cần cấu hình một số thông tin như ngôn ngữ, giao diện (font, color).







Đây là giao diện của Dev C++

