**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI: LẬP TRÌNH GAME AI LÀ TRIỆU PHÚ**

**BẰNG NGÔN NGỮ LẬP TRÌNH C**

Giảng viên hướng dẫn: TRẦN THỊ DUNG

Sinh viên thực hiện: TRẦN HUỲNH LƯU

ĐOÀN MINH PHÚC

Lớp: CQ.60.CNTT

Khoá:60

Tp. Hồ Chí Minh, năm 2020

**LỜI CẢM ƠN**

Sau gần 2 học kỳ học tập và rèn luyện tại bộ môn Công nghệ thông tin trường Đại học Giao thông Vận tải – Phân hiệu tại thành phố Hồ Chí Minh chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành bài tập lớn của mình.

Sau một thời gian nỗ lực thực hiện thì đề tài cũng đã hoàn thành. Nhưng không sao tránh khỏi những sai sót do chúng em còn chưa có nhiều kinh nghiệm thực tế. Em kính mong nhận được sự góp ý và nhận xét từ quý thầy, cô để em có thể hoàn thiện và hoàn thành tốt hơn cho đề tài của mình.

Lời sau cùng em một lần nữa kính chúc quý thầy, cô bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải – Phân hiệu tại thành phố Hồ Chí Minh thật nhiều sức khỏe và thành công.

*Tp. Hồ Chí Minh, ngày 18 tháng 6 năm 2020*

Sinh viên thực hiện

**Trần Huỳnh Lưu**

**Đoàn Minh Phúc**

**MỤC LỤC**

**Phần 1. Lập trình Game.....................................................................................1**

**Chương 1. Tổng quan về đề tài..........................................................................1**

**Chương 2. Xây dựng chương trình....................................................................3**

**Chương 3. Hướng dẫn sử dụng chương trình...................................................7**

**Chương 4. Giao diện............................................................................................8**

**Chương 5. Kết luận và đánh giá.........................................................................11**

**Phần 2. Lý thuyết.................................................................................................12**

**Chương 1. Làm việc với tệp................................................................................12**

**Chương 2. Danh sách liên kết đơn.....................................................................16**

**Chương 3. Thuật toán sắp xếp và tìm kiếm......................................................23**

**Tài liệu tham khảo...............................................................................................25**

**PHẦN 1. LẬP TRÌNH GAME**

**CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI**

**1.1 Lý do chọn đề tài**

Hiện nay, công nghệ ngày càng phát triển, cùng với đó là sự phát triển vượt trội của các phần mềm, ứng dụng, trong đó Facebook là một ứng dụng mà toàn cầu ai cũng biết đến. Gần đây thì Facebook có phát triển trò chơi trivia mang tên “Confetti”, trò chơi này mang thể loại tựa như game “Ai là triệu phú”, không những thách thức trí tuệ người chơi mà bên cạnh đó người chơi cũng có thể kiếm được tiền từ những trò chơi này. Confetti được thực hiện và phát triển ở một số quốc gia, trong đó có Việt Nam. Confetti Việt Nam diễn ra trong 2 mùa, sau 42 tuần phát sóng họ đã gửi lời chào tạm biệt đến khán giả, để lại cho nhiều bạn Việt những sự hối tiếc.

Ở Việt Nam còn có các app game trivia khác như VietNamIQ, TopIQ, Jingo,...hầu hết đều bắt nguồn từ tựa game trivia nổi tiếng của Mỹ mang tên HQTrivia.

Ngày nay, các game trivia ngày càng phát triển và tiếp cận với nhiều người hơn bởi những gì mà nó đem lại không những là những kiến thức mà còn có thể kiếm được tiền.

Ví như trong đại dịch Covid-19, nhiều người thất nghiệp, không thể kiếm tiền, thì sự phát triển các tựa game trivia này là sự cần thiết và mới mẻ, mọi người có thể chơi và kiếm tiền chỉ bằng một chiếc điện thoại cùng với sự hiểu biết của mình.

Những điều này đã thôi thúc cho chúng em đến với đề tài “Lập trình game ‘Ai là triệu phú’ bằng ngôn ngữ lập trình C” để tạo nên những bước đầu tiên để sau này có thể tạo các app trivia.

**1.2 Mục tiêu nghiên cứu:**

**-** Xây dựng chương trình game “Ai là triệu phú” dựa trên ngôn ngữ lập trình C

- Game phải đáp ứng được giao diện thân thiện với người dùng, dễ sử dụng.

- Sử dụng những câu hỏi mà có thể đem lại những kiến thức hay mà trước giờ nhiều người chưa biết hoặc nhầm lẫn.

**1.3 Phạm vi nghiên cứu:**

- Nghiên cứu về việc cài đặt thư viện đồ họa graphics.h

- Nghiên cứu và tìm hiểu về các hàm đồ họa trong thư viện graphics.h

- Nghiên cứu và tìm hiểu các câu hỏi hay.

- Nghiên cứu và tìm hiểu hàm bắt sự kiện chuột trong C

**CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH**

**2.1. Thuật toán:**

Game được xây dựng với thuật toán rất đơn giản, chủ yếu là sử dụng cấu trúc rẽ nhánh Switch case. Mỗi case từ 1-15 tương ứng với mỗi câu hỏi (có tổng cộng 15 câu). Nếu trả lời đúng thì sẽ đi đến case tiếp theo, còn sai thì sẽ break.

Đầu tiên là khai báo các thư viện, sau đó xây dựng các hàm nhỏ với chức năng khác nhau mà xuyên suốt trong các hàm đều có sử dụng các hàm đồ họa. (Các hàm đồ họa sẽ giải thích sau)

Hàm vẽ màn hình chính khi mới vào chương trình, sẽ hiện text “Who Want to Be a Millionaire of UTC2?” và text “Play Game”.

Hàm vẽ màn hình để chơi game, bao gồm 5 hình chữ nhật (1 dành cho câu hỏi và 4 dành cho đáp án) và 1 hình tròn dành cho “trợ giúp”.

Hàm cho phép dùng trợ giúp 50/50 từ câu 6 đến câu 10

Hàm trợ giúp, khi click vào trợ giúp thì sẽ thực hiện trợ giúp 50/50.

Hàm trả về kết quả chơi game, nếu trả lời sai bất kỳ câu nào thì sẽ thông báo việc endgame, còn trả lời đúng hết 15 câu thì thông báo chúc mừng.

Hàm hiển thị các câu hỏi, đáp án và số tiền hiện có, sử dụng cấu trúc switch case. Trong hàm này có sử dụng các hàm trợ giúp và hàm trả về kết quả chơi game

Hàm bắt đầu trò chơi, khi click chuột vào Play Game thì vào ngay câu số 1.

Hàm main, khởi động màn hình đồ họa, cài đặt chiều dài rộng màn hình đồ họa, khởi động hàm khung trang chủ và hàm chơi game.

**2.2. Các hàm đồ họa và một số hàm khác sử dụng trong chương trình:**

**2.2.1. Màu, nền, chữ:**

Đặt màu nền: setbkcolor(color); // color có thể là số hoặc chữ

Example: setbkcolor(0); hoặc setbkcolor(BLACK); đặt màu nền là màu đen

Chọn màu đường vẽ, màu chữ: setcolor(color);

Example: setcolor(4 ); hoặc setcolor(RED); đặt màu đường vẽ, màu in chữ trên màn hình đồ họa là màu đỏ

Chọn kiểu tô và màu tô: setfillstyle(kiểu tô, màu);

Example: setfillstyle(SOLID\_FILL,YELLOW); hoặc setfillstyle(1,14); chọn kiểu tô màu đặc và màu tô là màu vàng

Sử dụng Font chữ, hướng chữ và kích cỡ: settextstyle(font,hướng,kích cỡ);

Example: settextstyle(TRIPLEX\_FONT,0,6); chọn font Triplex, hướng chữ từ trái sang phải, kích cỡ lớn gấp 6 lần bình thường

In chữ ra màn hình đồ họa: outtextxy(tọa độ x, tọa độ y, “text”);

Example: outtextxy(0,0,Luudeptrai); in Luudeptrai từ vị trí có tọa độ(0,0).

Chọn kiểu đường cho các thủ tục vẽ đường line, rectangle,....:

setlinestyle(kiểu đường, mẫu, độ dày);

Example: setlinestyle(SOLID\_LINE,0,THICK\_WIDTH); kiểu đường nét liền, độ dày gấp ba lần

**2.2.2. Vẽ hình học:**

**Vẽ hình chữ nhật:**

Hình chữ nhật rỗng, chỉ có màu đường viền:

rectangle(x1,y1,x2,y2);

Trong đó x1,y1 là tọa độ điểm góc trái trên cùng của hcn và x2,y2 là tọa độ điểm góc phải dưới cùng của hcn

Example: rectangle(0,0,40,40);

Hình chữ nhật đặc, có tô màu:

bar(x1,x2,y1,y2);

Trong đó x1,y1 là tọa độ điểm góc trái trên cùng của hcn và x2,y2 là tọa độ điểm góc phải dưới cùng của hcn

Example: bar(20,10,50,80);

**Vẽ hình quạt:**

pieslice(x1,y1,a,b,r);

Trong đó x1,y1 là tọa độ tâm hình quạt; a là góc đầu, b là góc cuối, r là bán kính

Example: pieslice(50,70,0,360,20); vẽ hình tròn (vì góc a=0 và b=360), tâm (50,70), bán kính r=20

**2.2.3. Bắt sự kiện click chuột**

ismouseclick(WM\_LBUTTONDOWN): trả về true nếu xảy ra click chuột trái

getmouseclick(WM\_LBUTTONDOWN): lấy tọa độ mà ở đó có click chuột trái

Example: in từng tọa độ click chuột trái

while(1){ //\*Vòng lặp vô hạn\*//

if (ismouseclick(WM\_LBUTTONDOWN)){

getmouseclick(WM\_LBUTTONDOWN, x, y);

printf("left click : (%d,%d)\n", x, y);

}}

**2.2.4. Khởi động màn hình đồ họa**

int gd = DETECT, gm; //\*Sử dụng DETECT để chương trình tự chọn chế độ màn hình đồ họa phù hợp, lúc đó gm chúng ta không cần gán cho giá trị nào\*//

initgraph(&gd, &gm, ""); //\* “” là đường dẫn thư mục chứa file .BGI, nếu để trống thì chương trình sẽ tự tìm\*//

Kích thước màn hình đồ họa và đặt tên cho màn hình đồ họa khởi động:

initwindow(dai,rong,”text”);

Example: initwindow(1050,650,"Ai la trieu phu");

**CHƯƠNG 3. HƯỚNG DẪN SỬ DỤNG**

Chạy file “trieuphu.exe”, click vào “Play Game”, đọc câu hỏi và click chọn đáp án, nếu chọn sai sẽ hiển thông báo trả lời sai, nhấn Close ở góc phải bên trên màn hình game để thoát và chạy lại file “trieuphu.exe” nếu muốn chơi lại. Nếu trả lời đúng, chờ 2.5s để hiện câu hỏi tiếp.

Từ câu 6 đến câu 10 được sử dụng trợ giúp 50/50. Click vào trợ giúp, rồi click vào 50/50, 2 đáp án sai sẽ biến mất.

**CHƯƠNG 4. GIAO DIỆN**

**4.1. Giao diện màn hình ban đầu**



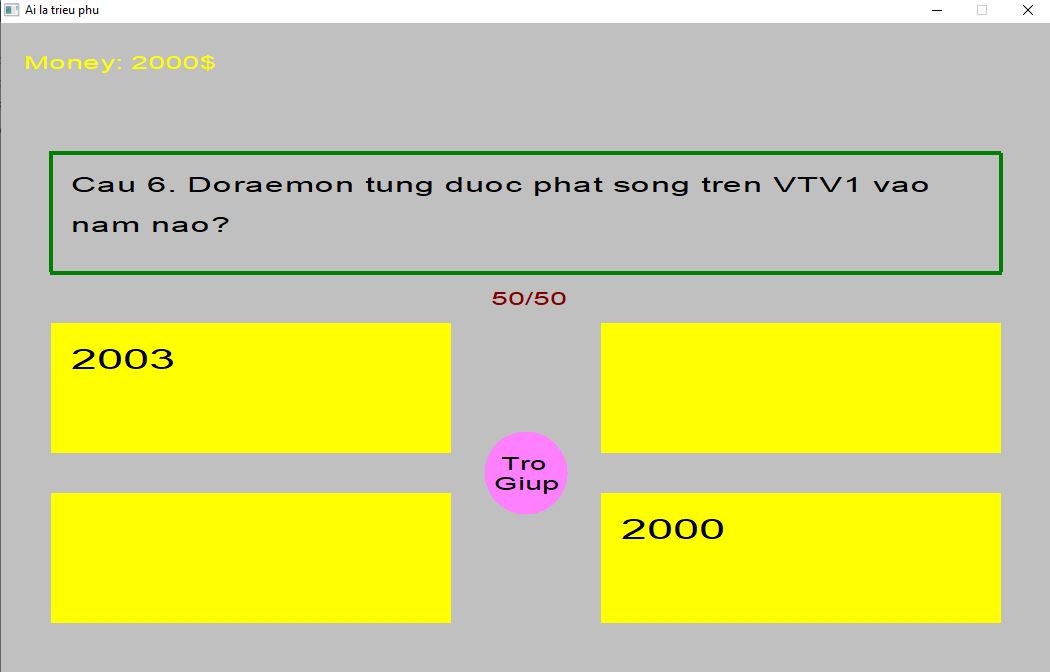
Hình 1.4.1. Giao diện màn hình bắt đầu, click vào Play Game để vào game

* 1. **Giao diện vào game**



Hình 1.4.2. Giao diện vào game, đọc câu hỏi và click vào các đáp án

* 1. **Giao diện trợ giúp 50/50**



Hình 1.4.3. Giao diện trợ giúp 50/50

* 1. **Giao diện thua game**



Hình 1.4.4. Giao diện khi trả lời sai

* 1. **Giao diện thắng game**



Hình 1.4.5. Giao diện khi trả lời đúng hết 15 câu

**CHƯƠNG 5: KẾT LUẬN VÀ ĐÁNH GIÁ**

* 1. **Ưu điểm**

- Giao diện game dễ nhìn.

- Người chơi có thể click chuột để chọn đáp án.

- Sử dụng nhiều câu hỏi có kiến thức hay và bất ngờ.

* 1. **Nhược điểm**

- Thuật toán còn khá đơn giản.

- Câu hỏi còn ít, chưa random được bộ đề.

- Chưa đưa ra những cách giải quyết vấn đề một cách tối ưu nhất trong khi code.

* 1. **Hướng phát triển**

- Tạo ra nhiều câu hỏi hơn và random bộ đề.

- Kết hợp với kiến thức làm việc với tệp để thuật toán câu hỏi và đáp án trở nên tốt hơn.

- Thêm tính năng Play game again.

**PHẦN 2. LÝ THUYẾT**

**CHƯƠNG 1 LÀM VIỆC VỚI TỆP**

1. **Tại sao chúng ta cần đến file?**

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

-Nếu chương trình của bạn có đầu vào(input) là lớn, bạn sẽ rất vất vả nếu phải nhập mỗi khi chạy. Thay vào đó, hãy lưu vào file và chương trình của bạn sẽ tự đọc mỗi lần khởi chạy

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

1. **Các kiểu file**

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

**2.2. 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.

1. **Thao tác với file trên ngôn ngữ C**
   1. **Khai báo sử dụng tệp**

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.

FILE biến\_con\_trỏ\_tệp;

Ví dụ: FILE \*taptin1

* 1. **Mở tệp**

Câu lệnh:

Biến con trỏ tệp = fopen(“file open”,”mode”);

Trong đó “file open” là đường dẫn dẫn tới file bạn cần mở, “mode” là tham số chúng ta cần chỉ định, mà các mode chúng ta hay sử dụng trong C như “w”, “wb”, “r”, “rb”.



Ví dụ: Mở 1 file .txt có tên “btl1”, chỉ cho phép đọc, có đường dẫn “C:\Users\PhongTran\Desktop”, sử dụng biến con trỏ tệp \*taptin1

taptin1 = fopen(“C:\\Users\\PhongTran\\Desktop”,”r”);

* 1. **Đóng tệp**

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

fclose(biến\_con\_trỏ\_tệp);

Ví dụ: fclose(taptin1);

* 1. **Đọc và ghi file**

Đọc file: sử dụng fscanf() đối với tệp .txt, sử dụng fread() đối với tệp nhị phân

Ghi file: sử dụng fprintf() đối với tệp .txt, sử dụng fwrite() đối với tệp nhị phân

* 1. **Ví dụ 1 đoạn code về đọc và ghi file.**

#include <stdio.h>

#include <stdlib.h>

int main()

{ int so; //\*Khai báo biến này để lưu giá trị số trong file btl1.txt\*//

FILE \*taptin1; //\*biến con trỏ này để đọc file\*//

FILE \*taptin2; //\*biến con trỏ này để ghi file\*//

taptin1=fopen("C:\\Users\\PhongTran\\Desktop\\btl1.txt","r");

//\*mở file btl1.txt và chỉ cho đọc\*//

taptin2=fopen("C:\\Users\\PhongTran\\Desktop\\btl2.txt","w");

//\*mở file btl2.txt và chỉ cho ghi\*//

if(taptin1==NULL) printf("Error\n");

//\*Trường hợp file btl1.txt không tồn tại thì sẽ báo lỗi\*//

fscanf(taptin1,"%d",&so);

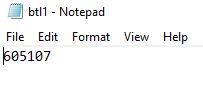
//\*đọc số trong file btl1.txt, trong file này đã có sẵn số 605107\*//

fprintf(taptin2,"%s","K60 CNTT UTC2 - BTL");

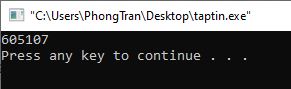
//\*ghi text “K60 CNTT UTC2 – BTL” vào file btl2.txt\*//

printf("%d\n",so); }

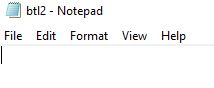
//\*in ra màn hình chương trình chạy số 605107 trong file btl1.txt\*//



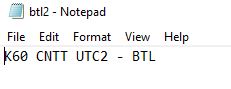
Hình 2.1.1 Tập tin btl1.txt ban đầu có số 605107.



Hình 2.1.2 Sau khi chạy chương trình, số 605107 được in ra



Hình 2.1.3 File btl2.txt trước khi chạy chương trình



Hình 2.1.4 File btl2.txt sau khi chạy chương trình

**CHƯƠNG 2 DANH SÁCH LIÊN KẾT ĐƠN**

1. **Danh sách liên kết đơn 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.



Hình 2.2.1 Hình ảnh mô tả cho một Node trong danh sách liên kết đơn



Hình 2.2.2 Hình ảnh mô phỏng một danh sách liên đơn kết đầy đủ

1. **Cài đặt danh sách liên kết đơn**
   1. **Khai báo danh sách liên kết**

Lấy ví dụ danh sách liên kết đơn lưu trữ các số nguyên.

typedef struct Lienketdon{

int value;

struct Lienketdon \*next;

} \*lkd;

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

Tại sao next lại là kiểu Lienketdon của chính nó? Bởi vì nó là con trỏ trỏ của chính bản thân nó, và nó trỏ tới một thằng Node kế tiếp cũng có kiểu Lienketdon.

* 1. **Tạo mới 1 Node**

lkd Taomoi(int giatri)

{

lkd tg;

tg=(lkd)malloc(sizeof(struct Lienketdon));

tg->next=NULL;

tg->value=giatri;

return tg;

}

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.

* 1. **Thêm node vào đầu**

lkd Themdau(lkd dau, int giatri)

{

lkd tg; tg=Taomoi(giatri);

if(dau==NULL)

{

dau=tg;

}

else

{

tg->next=dau;

dau=tg;

}

}

Việc thêm vào đầu đó là việc ta cập nhật lại giá trị đầu của danh sách lưu giữ giá trị. Một tg mang kiểu dữ liệu lkd được tạo mới và mang giá trị đầu là giatri, next sẽ là NULL. Chương trình kiểm tra danh sách cần thêm đầu đã có giá trị khác NULL chưa, nếu chưa thì nó sẽ được gán cho tg, nghĩa là nó sẽ mang giá trị là giatri, việc chèn đầu được hoàn tất . Nếu danh sách cần thêm đầu đã mang giá trị, thì giá trị cuối của tg là NULL sẽ gán cho giá trị đầu của danh sách cần chèn đầu, sau đó danh sách cần chèn đầu sẽ gán cho tg, như vậy việc chèn đầu được hoàn tất.

* 1. **Thêm node vào cuối**

lkd Themduoi(lkd duoi, int giatri)

{

lkd tg,p; tg=Taomoi(giatri);

if(duoi==NULL)

{

duoi=tg;

}

else

{

p=duoi;

while(p->next!=NULL){

p=p->next;

}

p->next=tg;

}

return duoi;

}

Khai báo tg mang kiểu dữ liệu lkd, tg được tạo mới mang một giá trị, giá trị đầu là giatri, next sẽ là NULL. Chương trình kiểm tra danh sách cần chèn đuôi đã mang giá trị khác NULL chưa, nếu chưa thì nó sẽ được gán cho tg, và lúc này giá trị của nó sẽ là giatri, việc chèn hoàn tất. Nếu danh sách cần chèn đã có giá trị khác NULL trước đó, khai báo p mang kiểu dữ liệu lkd, p được gán cho danh sách cần chèn, khi đó p sẽ mang giá trị của danh sách cần chèn. Câu lệnh while sẽ trỏ đến khi value của p bằng NULL, khi đó gán nó bằng tg, sau cùng return về danh sách cần chèn đuôi.

Sau đây là một đoạn code hoàn chỉnh về tạo danh sách liên kết đơn, thêm đầu và thêm đuôi

#include <stdio.h>

#include <stdlib.h>

typedef struct Lienketdon{

int value;

struct Lienketdon \*next;

} \*lkd;

lkd Taomoi(int giatri)

{

lkd tg;

tg=(lkd)malloc(sizeof(struct Lienketdon));

tg->next=NULL;

tg->value=giatri;

return tg;

}

lkd Themdau(lkd dau, int giatri)

{

lkd tg; tg=Taomoi(giatri);

if(dau==NULL)

{ dau=tg; }

else

{

tg->next=dau;

dau=tg;

}}

lkd Themduoi(lkd duoi, int giatri)

{

lkd tg,p; tg=Taomoi(giatri);

if(duoi==NULL)

{ duoi=tg; }

else

{

p=duoi;

while(p->next!=NULL){

p=p->next; }

p->next=tg;

}

return duoi;

}

lkd Nhap()

{ lkd nhap;

int n,so;

do{

printf("Nhap vao n\n"); scanf("%d",&n);

} while(n<=0);

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

{

printf("Nhap vao gia tri thu %d\n",i+1);

scanf("%d",&so);

if(i==0) {

nhap=Taomoi(so);

}

else nhap=Themduoi(nhap,so);

}

return nhap;

}

int main()

{

lkd vip=Nhap();

vip=Themdau(vip,7);

vip=Themduoi(vip,9);

for(lkd test=vip; test!=NULL; test=test->next)

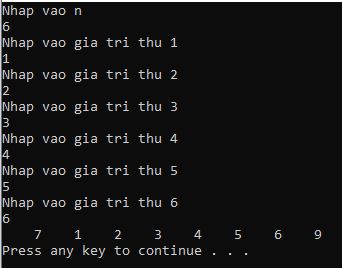
{

printf("%5d",test->value);

}

printf("\n");

}



Hình 2.2.3 Chương trình nhập 6 số 1,2,3,4,5,6 và thêm đầu số 7, thêm đuôi số 9

**CHƯƠNG 3 THUẬT TOÁN SẮP XẾP VÀ TÌM KIẾM**

**3.1. Thuật toán sắp xếp**

Ý tưởng là tìm từng phần tử cho mỗi vị trí của mảng hoán vị cần tìm.

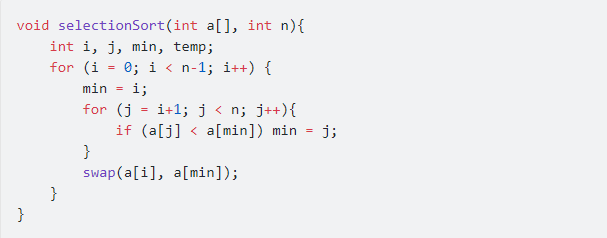
Thuật toán:

Tìm phần tử nhỏ nhất đưa vào vị trí 1

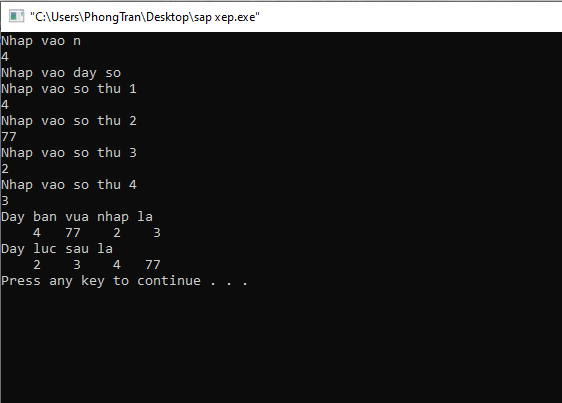
Tìm phần tử nhỏ tiếp theo đưa vào vị trí 2

Tìm phần tử nhỏ tiếp theo đưa vào vị trí 3

...



Hình 2.3.1 Hàm sắp xếp lựa chọn

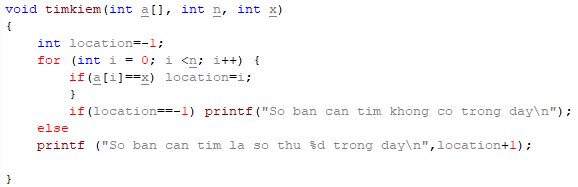


Hình 2.3.2 Chương trình chạy sắp xếp 1 dãy số trong 1 mảng vừa nhập

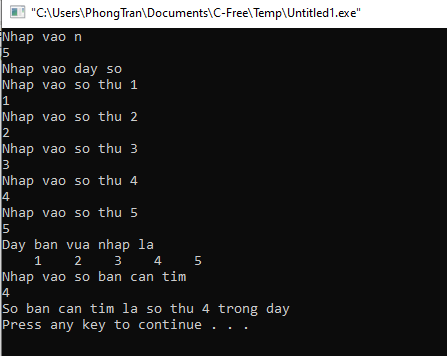
**3.2.** **Thuật toán tìm kiếm**

Một trong những thuật toán tìm kiếm cơ bản là tìm kiếm tuyến tính.

Tìm kiếm tuyến tính là kiểm tra tuần tự từng phần tử của mảng, đến khi nào giống thì thôi.



Hình 2.3.4 Hàm tìm kiếm



Hình 2.3.5 Chương trình chạy tìm kiếm 1 số trong 1 dãy số của 1 mảng

**TÀI LIỆU THAM KHẢO**

[1] Amy Lê, “Sử dụng graphics.h để xây dựng ứng dụng đồ họa”.

Available: https://www.stdio.vn/articles/su-dung-graphicsh-de-xay-dung-ung-dung-do-hoa-183

[2] Nguyễn Văn Quân, “[C/C++] Click chuột trong Dev-C++ – Mouse event in Dev-C++”

Available: https://nguyenvanquan7826.wordpress.com/2013/10/03/cc-click-chuot-trong-dev-c-mouse-event-in-dev-c/

[3] Nguyễn Hữu Tuấn, “Ngôn ngữ lập trình C – Đồ họa”

Available: https://voer.edu.vn/c/do-hoa/16cbe2bb/70d0d62d

[4] Nguyễn Văn Hiếu, “Danh sách liên kết đơn – Single linked list”

Available: https://nguyenvanhieu.vn/danh-sach-lien-ket-don/

[5] Nguyễn Văn Hiếu, “Đọc ghi file trong C”

Available: https://nguyenvanhieu.vn/doc-ghi-file-trong-c/