

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
KỸ THUẬT LẬP TRÌNH
ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ
KHÁCH SẠN

Giảng viên hướng dẫn : TRẦN THỊ DUNG

Sinh viên thực hiện : HOÀNG GIA KIẾT

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : K62

Thành phố Hồ Chí Minh, tháng 4 năm 2022

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
KỸ THUẬT LẬP TRÌNH
ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ KHÁCH SẠN

Giảng viên hướng dẫn : TRẦN THỊ DUNG

MSSV:

Sinh viên:

6251071049

HOÀNG GIA KIỆT

6251071068

NGUYỄN THỊ THANH NHƯ

6251071115

VÕ THỊ TƯỜNG VI

6251071078

LÊ TRUNG QUYỀN

Lớp

: Công nghệ thông tin

Khoá

: K62

Thành phố Hồ Chí Minh, tháng 4 năm 2022

GIỚI THIỆU SƠ LƯỢC

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Nhóm sinh viên thực hiện:

MSSV	Họ và tên
6251071049	: HOÀNG GIA KIẾT
6251071068	: NGUYỄN THỊ THANH NHƯ
6251071115	: VÕ THỊ TƯỜNG VI
6251071078	: LÊ TRUNG QUYỀN
Lớp	: CQ.62.CNTT

1. Tên đề tài

XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ KHÁCH SẠN

2. Mục đích, yêu cầu

a. Mục đích

Nhằm giúp các chủ khách sạn, cơ sở cho thuê phòng quản lý hiệu quả công việc. Công nghệ thông tin phát triển nên nhu cầu cần có một chương trình để tối ưu hóa hiệu suất, nâng cao hiệu quả làm việc, tiết kiệm thời gian, chi phí ghi chép bằng tay thủ công. Hệ thống được xây dựng trên nền tảng ứng dụng Desktop.

b. Yêu cầu

- Yêu cầu công nghệ
 - Sử dụng ngôn ngữ lập trình C.
 - Sử dụng danh sách liên kết đơn.
 - Sử dụng github.

- Yêu cầu
 - Chương trình có đầy đủ các chức năng như: thêm người thuê, tìm kiếm người thuê, chỉnh sửa, sắp xếp người thuê...
 - Ứng dụng cho phép người dùng: quản lý phòng và khách thuê,...
 - Giao diện: Thân thiện với người dùng và dễ dàng thao tác.

3. Nội dung và phạm vi đề tài

a. Nội dung

- Tổng quan bài toán.
- Tổng quan về các công nghệ đang sử dụng.
- Phân tích và thiết kế chương trình.
- Kiểm tra và chạy thử.
- Kết quả thu được

b. Phạm vi

- Lý thuyết về danh sách liên kết đơn
- Bài toán quản lý khách sạn

4. Công nghệ, công cụ và ngôn ngữ lập trình

- Sử dụng ngôn ngữ lập trình C.
- Sử dụng danh sách liên kết đơn
- Sử dụng github

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

Xây dựng thành công chương trình quản lý khách sạn đầy đủ các chức năng đã nêu trên.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: THS. TRẦN THỊ DUNG

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải

Điện thoại: 0388.389.579

Email: ttdung@st.utc2.edu.vn

LỜI CẢM ƠN

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến cô Dung trong **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải**. Những người đã truyền dạy, đã trang bị cho chúng em kho tàng kiến thức về bầu trời công nghệ thông tin rộng lớn.

Ở đây, chúng em không chỉ học được kiến thức về sách vở mà chúng em còn học được các bài học, kỹ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn. Đặc biệt, chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến cô Trần Thị Dung, người đã đồng hành cùng chúng em trong suốt quá trình làm đồ án, người đã bỏ thời gian quý báu, thậm chí là thời gian nghỉ ngơi để hướng dẫn, để định hướng đường đi nước bước cho chúng em. Chúng em thật chẳng biết dùng lời nào để diễn tả được công lao của cô. **“CẢM ƠN CÔ. CẢM ƠN CÔ VỀ TẤT CẢ”**.

Trong quá trình học tập và tìm hiểu chúng em đã nỗ lực rất nhiều với mong muốn hoàn thành bài tập lớn một cách tốt nhất, nhưng đời người sẽ có những thiếu sót không thể tránh khỏi, và với những người chưa chứng chạc và trưởng thành như chúng em thì sai lầm là không thể không mắc phải. Chúng em mong cô có thể thông cảm và cho chúng em những ý kiến, đóng góp để chúng em có thể hoàn thành Bài tập lớn của mình một cách trọn vẹn nhất.

Sau cùng, chúng em xin kính chúc Cô lời chúc sức khỏe, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Chúng em xin chân thành cảm ơn!

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên hướng dẫn

Trần Thị Dung

MỤC LỤC

GIỚI THIỆU SƠ LƯỢC	i
1. Tên đề tài.....	i
2. Mục đích, yêu cầu	i
a. Mục đích.....	i
b. Yêu cầu	i
3. Nội dung và phạm vi đề tài	ii
a. Nội dung.....	ii
b. Phạm vi	ii
4. Công nghệ, công cụ và ngôn ngữ lập trình	ii
5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng	ii
6. Giáo viên và cán bộ hướng dẫn.....	ii
LỜI CẢM ƠN	iii
NHẬN XÉT CỦA GIÁO VIÊN	iv
MỤC LỤC.....	v
CHƯƠNG 1: MỞ ĐẦU.....	1
1.1. Lý do chọn đề tài.....	1
1.2. Hướng tiếp cận của đề tài.....	1
1.3. Mục tiêu nghiên cứu.....	1
1.4. Đối tượng và phạm vi nghiên cứu.....	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	2
2.1. Danh sách liên kết đơn.....	2
2.1.1. Định nghĩa, tổng quan về danh sách liên kết đơn.....	2
2.1.2. Phân loại danh sách liên kết.....	2
2.1.3. Danh sách liên kết đơn.....	2
2.1.3.1. Khái niệm.....	2
2.1.3.2. Thao tác trên 1 node	3
2.1.3.3. Các thao tác trên danh sách liên kết đơn	3
2.2. Ngôn ngữ lập trình C	9

2.2.1. Giới thiệu	9
2.2.2. Khái niệm.....	9
2.2.3. Các đặc điểm của ngôn ngữ lập trình C.....	9
2.3. Github.....	11
2.3.1. Tổng quan về github	11
2.3.2. Khái niệm.....	11
2.3.3. Tính năng	11
2.3.4. Lợi ích gitbub mang lại cho lập trình viên.....	11
2.3.5. Các thuật ngữ cơ bản của github.....	12
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ	13
3.1. Đặc tả bài toán.....	13
3.2. Yêu cầu hệ thống.....	13
3.3. Phân tích và thiết kế chương trình	13
3.4. Cấu trúc dữ liệu chương trình	13
3.5. Các chức năng của chương trình.....	15
3.5.1. Thêm người thuê.....	15
3.5.2. Xem danh sách người thuê.....	17
3.5.3. Chỉnh sửa người thuê.....	18
3.5.4. Thanh toán	19
3.5.5. Xem tình trạng phòng	20
3.5.6. Sắp xếp người thuê theo số phòng.....	21
3.5.7. Tìm kiếm người thuê theo tên.....	22
3.5.8. In danh sách người thuê.....	23
3.5.9. Lưu thay đổi	24
CHƯƠNG 4: THỬ NGHIỆM ỨNG DỤNG	25
4.1. Giao diện đăng nhập	25
4.2. Giao diện sau khi đăng nhập.....	25
4.3. Giao diện thêm người thuê.....	26
4.4. Giao diện xem danh sách người thuê.....	26
4.5. Giao diện chỉnh sửa người thuê	27
4.6. Giao diện thanh toán	27

4.7. Giao diện xem tình trạng phòng.....	28
4.7.1. Giao diện menu sau khi chọn chức năng xem tình trạng phòng.....	28
4.7.2. Giao diện phòng trống	28
4.7.3. Giao diện phòng đầy	29
4.7.4. Giao diện tình trạng tất cả các phòng	29
4.8. Giao diện sắp xếp người thuê theo số phòng.....	29
4.9. Giao diện tìm kiếm người thuê theo tên	30
4.10. Giao diện in danh sách người thuê.....	30
4.10.1. File sau khi in.....	31
4.11. Giao diện lưu thay đổi.....	31
4.12. Giao diện sau khi thoát.....	31
CHƯƠNG 5 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	32
5.1. Kết luận	32
5.1.1. Kết quả đạt được	32
5.1.2. Kiến nghị.....	32
5.2. Hướng phát triển	32
TÀI LIỆU THAM KHẢO	33

CHƯƠNG 1: MỞ ĐẦU

1.1. Lý do chọn đề tài

Chúng ta, thế hệ đang sống trong thời đại công nghiệp hóa, hiện đại hóa. Thời đại mà công nghệ thông tin đang ngày một phát triển như vũ bão. Nó có mặt hầu hết trong tất cả các lĩnh vực mà chúng ta đang sinh hoạt, làm việc và cũng như giải trí....Nó giúp đỡ cho chúng ta từ nông nghiệp, công nghiệp, kinh tế, giáo dục, y tế.

Chính vì thế chương trình quản lý khách sạn là một giải pháp tối ưu nhằm:

- Tiết kiệm thời gian tối đa
- Tiết kiệm chi phí
- Quản lý hiệu quả, khoa học
- Tạo ra sự tiện ích cho chủ khách sạn

1.2. Hướng tiếp cận của đề tài

- Nắm vững kiến thức nền tảng của ngôn ngữ lập trình C, tiến đến phát triển giao diện.
- củng cố lại kiến thức về danh sách liên kết đơn
- Tiến hành kiểm tra và chạy thử phần mềm

1.3. Mục tiêu nghiên cứu

- Phát triển thành công chương trình quản lý khách sạn.
- Khiến ứng dụng trở nên có ích hơn cho xã hội.

1.4. Đối tượng và phạm vi nghiên cứu

- Đối tượng : Các chủ phòng khách sạn, dãy khách sạn .
- Phạm vi ứng dụng : Mong muốn phần mềm có thể có mặt hầu hết trong các phòng khách sạn, dãy khách sạn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Danh sách liên kết đơn

2.1.1. Định nghĩa, tổng quan về danh sách liên kết đơn

Danh sách liên kết là danh sách mà các phần tử (Node) liên kết với nhau nhờ vào vùng liên kết của chúng. Mỗi node bao gồm 2 phần : Phần Data dùng để chứa dữ liệu cần xử lý và phần liên kết dùng để liên kết tới các node khác.

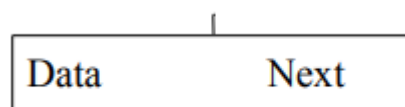
2.1.2. Phân loại danh sách liên kết

- Tùy cách liên kết giữa các phần tử, danh sách liên kết gồm có nhiều loại khác nhau :
 - Danh sách liên kết đơn: mỗi phần tử liên kết với phần tử đứng sau nó trong danh sách
 - Danh sách liên kết đôi/kép: mỗi phần tử liên kết với các phần tử đứng trước và sau nó trong danh sách.
 - Danh sách liên kết vòng: phần tử cuối danh sách liên kết với phần tử đầu danh sách.

2.1.3. Danh sách liên kết đơn

2.1.3.1. Khái niệm

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 đến NULL, nghĩa là đó là phần tử cuối cùng của link list. Các phần tử trong danh sách không thể truy cập trực tiếp, muốn truy cập bất kỳ phần tử nào đó trong danh sách phải xuất phát từ phần tử đầu tiên. Mỗi Node trong danh sách có thể được biểu diễn như sau:



Cú pháp:

```
Struct Node
{
    Kiểu_Dữ_Liệu Data;
    Node *Next;
};
```

Có thể hình dung danh sách liên kết qua hình vẽ như sau:



2.1.3.2. Thao tác trên 1 node

- Khai báo biến con trỏ p là biến con trỏ kiểu Node : Node *p
- Cấp phát bộ nhớ cho Node mới – được trỏ bởi con trỏ p :

- malloc

```
type *p;
p=(type *) malloc (byte_size);
```

- calloc

```
type *p;
p=(type *) calloc (n, byte_size);
```

- realloc

```
type *p;
P=realloc(p, n);
```

- Truy cập vào trường info : p->info
- Truy cập vào trường next : p->next
- Xóa Node được trỏ bởi p : free(p)

2.1.3.3. Các thao tác trên danh sách liên kết đơn

Thao tác trên danh sách liên kết khác với thao tác trên mảng khi cần thêm Node vào danh sách liên kết thì ta xin cấp phát bộ nhớ cho Node và nối Node đó vào danh sách liên kết và khi không cần sử dụng thì ta giải

phóng nó khỏi danh sách. Truy cập trên danh sách cũng khác so với trên mảng. Nếu ở mảng ta truy cập trực tiếp thông qua chỉ số mảng (thông qua địa chỉ ô nhớ) thì ở danh sách liên kết đơn ta phải truy cập một cách tuần tự. Khi làm việc trên danh sách liên kết đơn cũng tương tự như vậy nên ta cần chú ý những điểm sau:

Danh sách luôn có con trỏ đầu danh sách.

Danh sách luôn có giá trị báo kết thúc danh sách.

Trường Next của mỗi Node chỉ chứa địa chỉ của Node sau đó, trừ Node cuối.

Không tách danh sách thành 2 danh sách con nếu danh sách phần sau chưa có con trỏ tới.

Khởi tạo danh sách rỗng.

- **Thêm một phần tử (node) vào danh sách:**

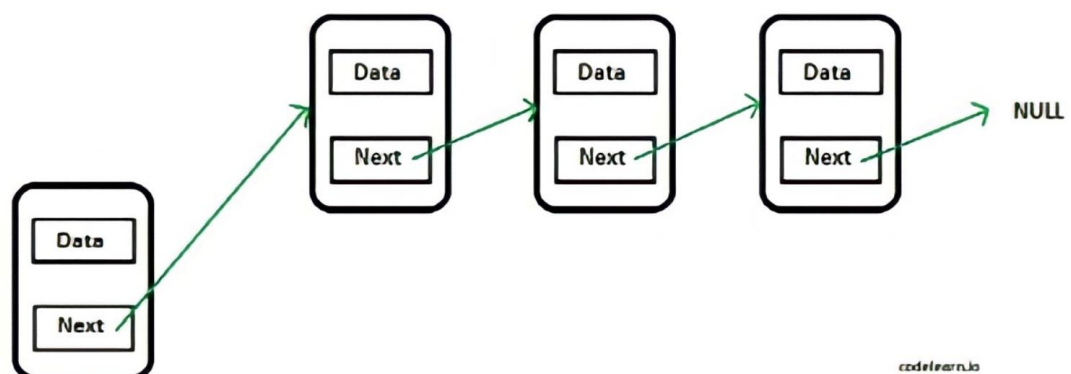
- Trường hợp 1 : Thêm phần tử vào đầu danh sách :

+ Bước 1 : Cho con trỏ của node cần thêm (node p) liên kết với node đầu danh sách pHead.

$p \rightarrow pNext = l.pHead;$

+ Bước 2 : Cập nhập lại node đầu pHead là node vừa thêm (node p).

$l.pHead = p;$



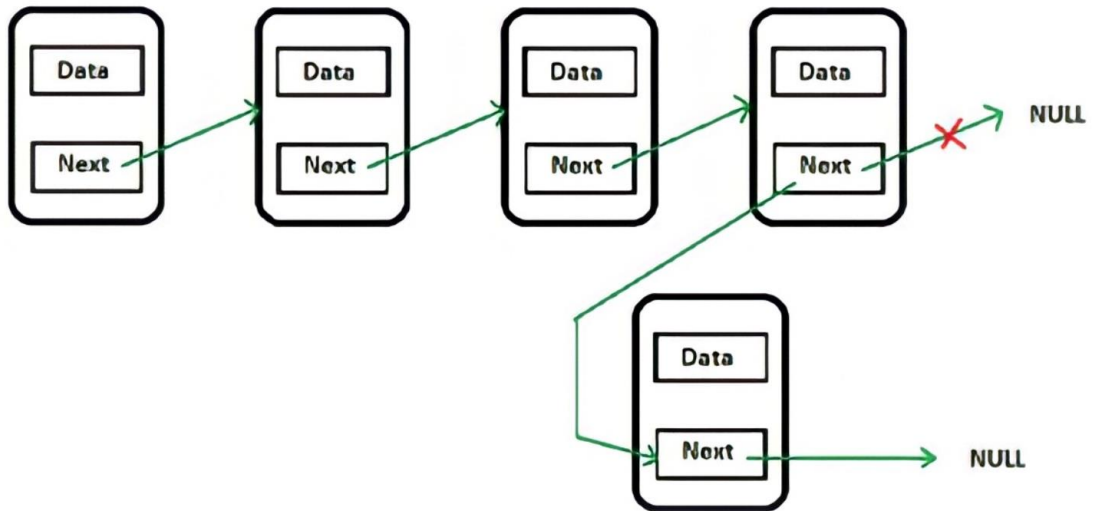
- Trường hợp 2 : Thêm phần tử vào cuối danh sách :

+ Bước 1 : Cho con trỏ của node cuối danh sách pTail liên kết với node cần thêm (node p).

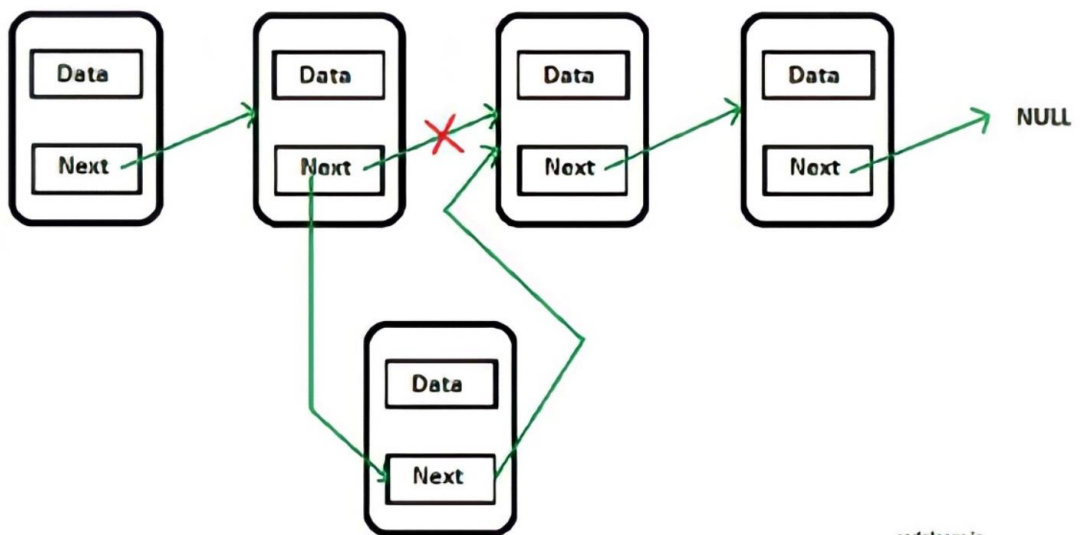
l.pTail->pNext=p;

+ Bước 2 : Cập nhật lại node cuối pTail chính là node mới thêm (node p).

l.pTail=p;



- Trường hợp 3 : Thêm phần tử vào vị trí bất kì.



codelearn.io

- Xóa một phần tử ra khỏi danh sách :

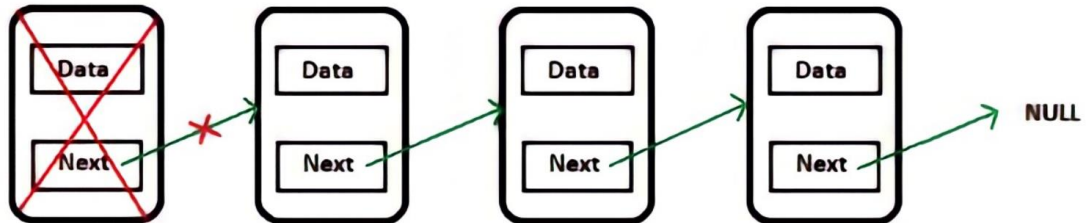
- Trường hợp 1 : Xóa phần tử (p) đầu danh sách :

+ Bước 1 : Cho con trỏ node đầu danh sách (pHead) trỏ đến node sau nó (pNext).

```
l.pHead=l.pHead->pNext;
```

+ Bước 2 : Xóa đi node (p) nằm ở đầu danh sách.

```
free(p);
```



- Trường hợp 2 : Xóa phần tử cuối danh sách :

+ Bước 1 : Nếu danh sách có một phần tử thực hiện hàm xóa đầu (Trường hợp 1).

+ Bước 2 : Nếu danh sách có nhiều phần tử ta thực hiện duyệt danh sách từ đầu đến kế cuối.

```
for(NODE *k=l.pHead; k!=NULL; k=k->pNext)
```

+ Bước 3 : Phát hiện phần tử cần xóa ở vị trí kế cuối.

```
if(k->pNext==l.pTail)
```

+ Bước 4 : Xóa đi phần tử cuối.

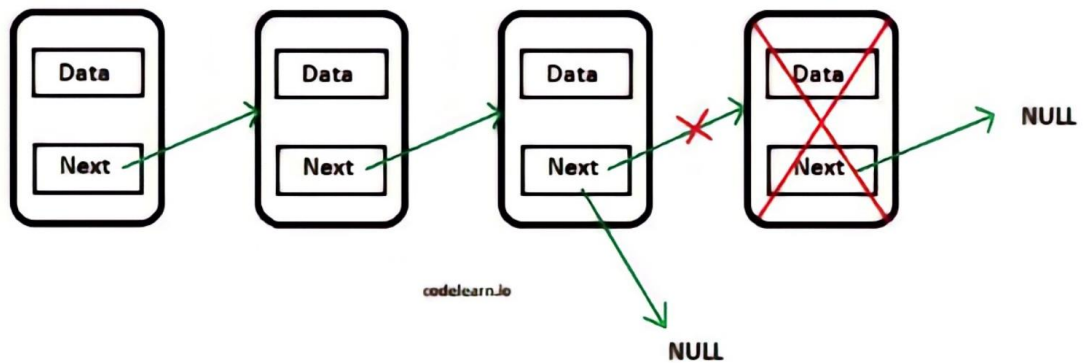
```
Free(l.pTail);
```

+ Bước 5 : Cho con trỏ kế cuối trỏ đến NULL.

```
k->pNext=NULL;
```

+ Bước 6 : Cập nhập lại pTail.

```
l.pTail=k;
```



- Trường hợp 3 : Xóa phần tử ở vị trí bất kì trong danh sách :

+ Bước 1 : Nếu node cần xóa nằm đầu danh sách thì xóa đầu (Trường hợp 1).

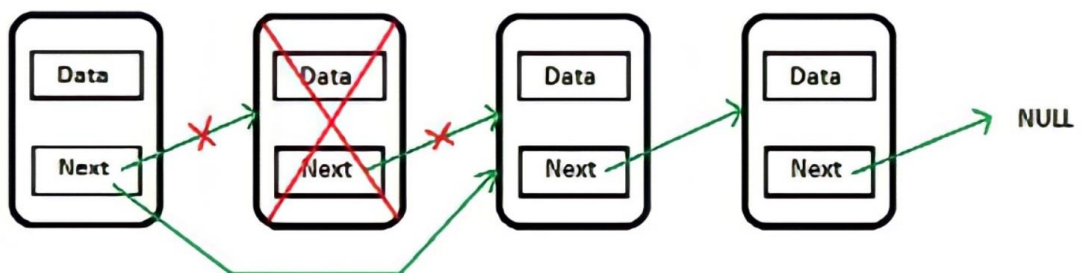
+ Bước 2 : Nếu node cần xóa nằm cuối danh sách thì xóa cuối (Trường hợp 2).

+ Bước 3 : Nếu node cần xóa không nằm ở đầu hoặc cuối danh sách :

Duyệt danh sách liên kết đơn.

Phát hiện node cần xóa ta cập nhật mỗi liên kết của node trước node cần xóa với node sau node cần xóa.

Xóa phần tử cần xóa.



- Tìm kiếm phần tử trong danh sách :

+ Bước 1: Cho con trỏ p trở vào đầu danh sách liên kết (l) và khởi tạo 1 biến int found=0

```
Node *p=l.pHead;
```

```
int found=0;
```


+ Bước 2: Thực hiện duyệt các phần tử từ đầu đến cuối danh sách kiểm tra nếu có **data= X** thì trả về địa chỉ của Node đó, ngược lại thì cho con trỏ p trở về vị trí phần tử kế tiếp

```
while ((p!=NULL)&&(p->data!=x))  
{  
    p=p->pNext;  
}  
return p;
```

2.2. Ngôn ngữ lập trình C

2.2.1. Giới thiệu

C là một ngôn ngữ lập trình được phát triển bởi Dennis M. Ritchie để phát triển hệ điều hành UNIX và được thực thi đầu tiên trên máy tính DEC PDP-11. Năm 1987 hãng Borland đã cung cấp phần mềm Turbo C-làm cho việc sử dụng ngôn ngữ C trở nên dễ dàng. Trong quá trình phát triển đã xuất hiện nhiều phiên bản C khác nhau. Để tiến tới thống nhất 1990 ANSI đã xây dựng ngôn ngữ C chuẩn.

2.2.2. Khái niệm

C là một trong ngôn ngữ lập trình cấu trúc tiêu biểu có tốc độ xử lý nhanh, có công cụ mạnh để quản lý bộ nhớ và các thiết bị ngoại vi. C có cấu trúc độc lập và được sử dụng rộng rãi để viết các ứng dụng, hệ điều hành như Windown và nhiều chương trình phức tạp khác nhau Oracle database, Git, Python Interpreter,...

2.2.3. Các đặc điểm của ngôn ngữ lập trình C

Tính cô đọng (compact) : C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết được biểu diễn bằng những chuỗi kí tự ngắn gọn.

Tính cấu trúc (structured) : C có một tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng dễ hiểu.

Tính tương thích (compatible) : C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình được viết bằng C vẫn hoàn toàn tương thích.

Tính linh động (flexible) : C là một ngôn ngữ uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.

Biên dịch (compile) : C cho phép biên dịch nhiều tập tin thành chương trình riêng rẽ thành các tập tin đối tượng (object) và liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

Con trỏ: C cung cấp tính năng con trỏ, chúng ta có thể tương tác trực tiếp với bộ nhớ bằng việc sử dụng con trỏ và có thể sử dụng con trỏ cho bộ nhớ, cấu trúc, hàm, mảng,....

Độ quy: Trong C, chúng ta có thể gọi hàm ngay bên trong bản thân đó. Nó cung cấp mã có khả năng sử dụng lại cho mỗi hàm.

Ngôn ngữ lập trình có cấu trúc: C là ngôn ngữ lập trình theo nghĩa chúng ta chia chương trình thành các phần bằng cách sử dụng hàm, vì thế nó dễ hiểu và dễ chỉnh sửa và vì sử dụng hàm nên code C có khả năng tái sử dụng.

2.3. Github

2.3.1. Tổng quan về github

GitHub được viết bằng Ruby on Rails và Erlang do Tom Preston-Werner, Chris Wanstrath, và PJ Hyett phát triển trang web được đưa ra và chạy chính thức vào tháng 4 năm 2008.

Tính đến thời điểm tháng 3 năm 2018 Github đang là dịch vụ máy chủ lưu trữ các mã nguồn lập trình lớn nhất thế giới. Với hơn 25 triệu người dùng và hơn 80 triệu mã nguồn dự án, Github đã trở thành một phần không thể thiếu đối với cộng đồng phát triển mã nguồn mở và cộng đồng lập trình viên trên toàn thế giới.

2.3.2. Khái niệm

GitHub là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau. Vài thông tin về GIT : Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.

2.3.3. Tính năng

Follow User : tính năng này giúp bạn có thể theo dõi những hoạt động của người khác.

Watch Project : đây là tính năng giúp mọi người có thể theo dõi được các hoạt động của nhiều chương trình khác nhau. Nhờ đó họ có thể hiểu xem quá trình để mọi người phát triển các phần mềm như thế nào, hoạt động nhờ project này ra làm sao.

Wiki, thống kê, issue... được đặt vào các namespace và coi là các user.

2.3.4. Lợi ích gitbub mang lại cho lập trình viên

Để dàng Markdown: bạn có thể chỉnh sửa được các cách hiển thị của document, ví dụ như định dạng in nghiêng, in đậm, tạo list, thêm hình...

Có thể tracking về sự thay đổi qua nhiều version: nếu bạn có nhiều member cùng thực hiện một dự án, sử dụng Github sẽ giúp bạn lưu lại thay

đổi mà bạn đã đẩy lên repository. Có thể nói rằng chức năng này khá giống với Google Drive.

Quản lý các source code cực kỳ dễ dàng: nếu bạn muốn tạo ra repo, các nguồn code từ repo đó đều đã được lưu sẵn ở trên Github. Nó giúp bạn có thể coi lại quá trình mà mình đã làm việc thông qua những bình luận sau mỗi lần mà bạn commit. Trên thực tế, nhiều người có thể cùng làm một repo chứ không phải chỉ một người.

2.3.5. Các thuật ngữ cơ bản của github

git init: tạo một repo trên máy của bạn.

git add: cập nhập lại project.

git clone: sao chép một repo trên server về máy.

git commit: xác nhận và lưu những thay đổi của project.

git push: cập nhập những thay đổi của project và gửi lên server.

git checkout: tạo một nhánh mới và chuyển qua nhánh đó hoặc chuyển nhánh.

git remote: Xem danh sách remote, kết nối repo của bạn đến máy chủ remote, đổi tên remote.

git branch: tạo nhánh mới, xóa nhánh, kiểm tra nhánh hiện tại.

git pull: lấy source mới nhất trên server về và tiến hành trộn.

git merge: để trộn một nhánh vào nhánh đang hoạt động.

git log: hiển thị danh sách các commit trên một nhánh và các thông tin của nó.

git diff: xem xung đột với tệp cơ sở, với nhánh được trộn trước khi trộn.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ

3.1. Đặc tả bài toán.

Ngày nay khi mà khoa học kỹ thuật phát triển, đặc biệt là sự bùng nổ công nghệ thông tin thì việc quản lý khách sạn sẽ dễ dàng hơn nhiều. Với sự phát triển mạnh của công nghệ thông tin như hiện nay dẫn đến nhu cầu xây dựng chương trình quản lý khách sạn. Yêu cầu của bài toán là tạo ra chương trình có thể thực hiện các thao tác quản lý khách sạn một cách dễ dàng, tiện lợi dựa trên sự trợ giúp của máy tính. Mọi công việc phải được thao tác trên một vùng dữ liệu chung để đảm bảo việc đồng bộ với nhau trong khâu quản lý.

Một người cần quản lý thông tin về các phòng khách sạn của mình. Mỗi phòng gồm các thuộc tính: số phòng, loại phòng, tình trạng phòng. Với mỗi khách hàng thuê phòng sẽ có các thuộc tính của khách hàng : họ tên, ngày sinh, giới tính, số điện thoại, căn cước công dân.

3.2. Yêu cầu hệ thống

Quản lý khách sạn là chương trình quản lý hồ sơ của khách thuê phòng trong quá trình thuê phòng ở khách sạn. Chương trình có thể thực hiện các công việc : thêm người thuê, xem danh sách người thuê, chỉnh sửa người thuê, thanh toán, xem tình trạng phòng, sắp xếp danh sách tăng dần theo số phòng, tìm kiếm người thuê theo tên, in danh sách người thuê. Chương trình được viết bằng ngôn ngữ C và dựa trên cấu trúc lưu trữ của danh sách liên kết đơn .

3.3. Phân tích và thiết kế chương trình

3.4. Cấu trúc dữ liệu chương trình

Để chương trình quản lý khách sạn được chắc chắn thì mỗi khách hàng, cũng như mỗi phòng thuê có thuộc tính như sau :

- Khách thuê phòng :
 - Họ và tên: sử dụng kiểu dữ liệu char, với độ dài tối đa 50 kí tự, có thể trùng lặp.

- Ngày tháng năm sinh: sử dụng struct Date để nhập ngày tháng năm sinh, có thể trùng lặp.

```
struct Date
{
    int day, month, year;
};
```

- Giới tính: sử dụng kiểu dữ liệu bool để lấy giới tính khách thuê, 0: nữ, 1: nam.
- Số điện thoại: sử dụng kiểu dữ liệu char, với độ dài 15 ký tự, mỗi khách hàng có duy nhất một số điện thoại không được trùng lặp.
- Căn cước công dân: sử dụng kiểu dữ liệu char, với độ dài tối đa 15 ký tự, mỗi khách hàng có một số căn cước công dân riêng, không được trùng lặp.

```
struct RoomRenter
{
    char fullName[50];           // họ tên
    Date birthDay;               // ngày tháng năm sinh
    bool sex;                    // 0:nữ 1:nam
    char phoneNumber[15];        // số điện thoại
    char identityCard[15];       // CCCD
    Room *room = NULL;          // trỏ đến phòng thuê
};
```

- Phòng cho thuê :

- Số phòng: sử dụng kiểu dữ liệu int, mỗi phòng có một số phòng khác nhau để phân biệt các phòng
- Loại phòng: sử dụng kiểu dữ liệu char, với độ dài tối đa 10 ký tự, được phép trùng lặp
- Tình trạng phòng: sử dụng kiểu dữ liệu bool, 0: phòng trống, 1: phòng đầy.

```
struct Room
{
    int number;           // số phòng
    char type[10];        // loại phòng: thường or cao cấp
    bool isActive;        // tình trạng phòng - 0:trống 1:đầy
};
```

3.5. Các chức năng của chương trình

3.5.1. Thêm người thuê

- Cho nhập vào thông tin người thuê.
- Kiểm tra số điện thoại, căn cước có trùng lặp, hoặc không hợp lệ.
- Nếu tất cả đã hợp lệ, cập nhập tình trạng phòng về phòng đầy, xuất ra thông tin người vừa thuê.
- Thêm người thuê vào cuối danh sách.


```

void addARoomRenter(RoomRenters &roomRenters, Room rooms[], int n)
{
    RoomRenter r;
    inputARoomRenter(r, rooms, n);
    do
    {
        If (isExitPhoneNumber(roomRenters, r.phoneNumber)){
            system("cls");
            printf("\n\t%40c(!) Da ton tai nguoi thue voi so dien thoai nay (!)\n\a", '
');
            printf("\n%50c(?) Nhap lai so dien thoai khac: ", ' ');
            scanf("%s", &r.phoneNumber);
        }
        // nhập lại mà ko hợp lệ - nhập lại tiếp
        do
        {
            if (strlen(r.phoneNumber) <= 0 || strlen(r.phoneNumber) > 10 ||
strlen(r.phoneNumber) != 10)
            {
                system("cls");
                printf("\n\t%40c(!) So dien thoai khong hop le - Nhap lai
(!)\n\a", ' ');
                printf("\n%50c(?) Nhap so dien thoai (10 so): ", ' ');
                scanf("%s", &r.phoneNumber);
            }
        } while (strlen(r.phoneNumber) <= 0 || strlen(r.phoneNumber) > 10 ||
strlen(r.phoneNumber) != 10);
        system("cls");
    } while (isExitPhoneNumber(roomRenters, r.phoneNumber));
    Do {
        if (isExitIdentityCard(roomRenters, r.identityCard)){
            system("cls");
            printf("\n\t%40c(!) Da ton tai nguoi thue voi so CCCD nay (!)\n\a",
' ');
            printf("\n%50c(?) Nhap lai so CCCD khac: ", ' ');
            scanf("%s", &r.identityCard);
        }
        do
        {
            if ((strlen(r.identityCard) <= 0 || strlen(r.identityCard) > 12) ||
strlen(r.identityCard) != 12)
            {
                system("cls");
                printf("\n\t%40c(!) So CCCD khong hop le - Nhap lai (!)\n\a", '
');
                printf("\n%50c(?) Nhap so CCCD (12 so): ", ' ');
                scanf("%s", &r.identityCard);
            }
        } while ((strlen(r.identityCard) <= 0 || strlen(r.identityCard) > 12)
|| strlen(r.identityCard) != 12);
        system("cls");
    } while (isExitIdentityCard(roomRenters, r.identityCard));
    r.room->isActive = 1; // cập nhật tình trạng phòng -> 1 (phòng đầy)
    system("cls");
    printf("\n\t%50cTHONG TIN NGUOI THUE VUA NHAP\n", ' ');
    outputARoomRenterByVertical(r);
    addNodeInTail(roomRenters, r);
}

```

3.5.2. Xem danh sách người thuê

- Xây dựng hàm xuất thông tin người thuê theo chiều ngang.
- Duyệt từ đầu đến cuối danh sách, và xuất danh sách người thuê theo chiều ngang.

```
void outputRoomRenters(RoomRenters roomRenters)
{
    int count = 1;
    printf("+ ----- + ----- + ----- + ----- + \n");
    printf("|      STT       |              Ho va ten              |      Ngay\nsinh   | Gioi tinh   | So dien thoai   |          So CCCD           | So\phong | \n");
    printf("+ ----- + ----- + ----- + ----- + \n");
    for (Node *t = roomRenters.pHead; t != NULL; t = t->pNext)
    {
        printf("| %-7d ", count++);
        outputARoomRenterByHorizontal(t->data);
    }
    printf("+ ----- + ----- + ----- + ----- + \n");
}
```

3.5.3. Chỉnh sửa người thuê

- Xây dựng hàm để lựa chọn tiêu chí cần chỉnh sửa.
- Cho nhập vào số điện thoại để tìm người cần chỉnh sửa và kiểm tra xem có tồn tại người thuê không.
- Gọi hàm lựa chọn tiêu chí chỉnh sửa vừa xây dựng để người thuê chọn tiêu chí muốn sửa.
- Xuất ra thông tin mới sau khi đã chỉnh sửa.

```
void updateRoomRenterV2(RoomRenters &roomRenters, Room rooms[],
int n, char *phoneNumber)
{
    if (!isExitPhoneNumber(roomRenters, phoneNumber))
        printf("\n\t%40c(!) Không tồn tại người thuê này (!)\n\a",
        ' ');
    else
    {
        for (Node *t = roomRenters.pHead; t != NULL; t = t->pNext)
        {
            if (strcmp(t->data.phoneNumber, phoneNumber) == 0)
            {
                chooseFunctionToUpdate(t->data, rooms, n);
                system("cls");
                printf("\n\t%40c(*) Chỉnh sửa thành công (*)\n", ' ');
                break;
            }
        }
    }
}
```

3.5.4. Thanh toán

- Xoá người thuê

```
void deleteARoomRenter(RoomRenters &roomRenters, char
*phoneNumber)
{
    if (strcmp(roomRenters.pHead->data.phoneNumber, phoneNumber)
== 0)
    {
        roomRenters.pHead->data.room->isActive = 0;
        removeNodeInHead(roomRenters);
        printf("\n\t%40c(*) Thanh toan thanh cong (*)\n", ' ');
    }
    else if (strcmp(roomRenters.pTail->data.phoneNumber,
phoneNumber) == 0){
        roomRenters.pTail->data.room->isActive = 0;
        removeNodeInTail(roomRenters);
        printf("\n\t%40c(*) Thanh toan thanh cong (*)\n", ' ');
    }
    else{
        if (!isExitPhoneNumber(roomRenters, phoneNumber))
            printf("\n\t%40c(!) Khong ton tai nguoi thue nay
(!)\n\ta", ' ');
        else
        {
            Node *g = NULL; // node nằm trước node cần xoá
            for (Node *t = roomRenters.pHead; t != NULL; t = t-
>pNext)
            {
                if (strcmp(t->data.phoneNumber, phoneNumber) ==
0){
                    t->data.room->isActive = 0;
                    g->pNext = t->pNext;
                    free(t);
                    printf("\n\t%40c(*) Thanh toan thanh cong
(*)\n", ' ');
                    return;
                }
                g = t;
            }
        }
    }
}
```

3.5.5. Xem tình trạng phòng

- Duyệt qua danh sách người thuê.
- Kiểm tra tình trạng phòng.
- Xuất ra danh sách các phòng đầy, trống, tất cả các phòng.

```
case 5:
{
    int choose;
    bool exit = false;
    do
    {
        system("cls");
        printf("%50c#####\n", ' ');
        printf("%50c##          TINH TRANG PHONG          ##\n", ' ');
        printf("%50c##-----##\n", ' ');
        printf("%50c##          1. Phong trong          ##\n", ' ');
        printf("%50c##          2. Phong day          ##\n", ' ');
        printf("%50c##          3. Tat ca          ##\n", ' ');
        printf("%50c##-----##\n", ' ');
        printf("%50c##          0. Tro ve          ##\n", ' ');
        printf("%50c#####\n", ' ');
        printf("%50c    Lua chon cua ban -> ", ' ');
        scanf("%d", &choose);
        switch (choose)
        {
            case 1:
            {
                system("cls");
                printf("\n\t%50c1. PHONG TRONG\n", ' ');
                outputRoomsAreEmpty(rooms, n);
                system("pause");
                break;
            }
            case 2:
            {
                system("cls");
                printf("\n\t%50c2. PHONG DAY\n", ' ');
                outputRoomsAreWorking(rooms, n);
                system("pause");
                break;
            }
            case 3:
            {
                system("cls");
                printf("\n\t%50c3. TAT CA\n", ' ');
                outputRooms(rooms, n);
                system("pause");
                break;
            }
            case 0:
            {
                exit = true;
                break;
            }
            default:
            {
                printf("\n\t%50c(!) Lua chon khong hop le (!)\n\ta", ' ');
                system("pause");
                break;
            }
        }
    } while (!exit);
    break;
}
```

3.5.6. Sắp xếp người thuê theo số phòng

- Duyệt qua danh sách người thuê.
- Sắp xếp người thuê theo số phòng.
- Xuất ra danh sách người thuê.

```
void sortRoomRentersAscendingByRoomNumber(RoomRenters
&roomRenters)
{
    for (Node *t = roomRenters.pHead; t != NULL; t = t->pNext)
    {
        Node *minIndex = t;
        for (Node *g = t->pNext; g != NULL; g = g->pNext)
        {
            if (minIndex->data.room->number > g->data.room-
>number)
                minIndex = g;
        }
        if (minIndex != t)
            swapTwoRoomRenters(t->data, minIndex->data);
    }
}
```

3.5.7. Tìm kiếm người thuê theo tên

- Duyệt qua danh sách người thuê.
- Nhập tên người thuê để tìm kiếm.
- Xuất ra những người thuê có các kí tự trùng với kí tự nhập vào tìm kiếm.

```
void findRoomRentersByName(RoomRenters roomRenters, char *name)
{
    int count = 1;
    printf("+ ----- + ----- + ----- + ----- + \n");
    printf("|      STT       |              Ho va ten              |      Ngay\nsinh   | Gioi tinh | So dien thoai |          So CCCD           | So\nphong  | \n");
    printf("+ ----- + ----- + ----- + ----- + \n");
    for (Node *t = roomRenters.pHead; t != NULL; t = t->pNext)
    {
        // dùng strlwr() để chuyển tên về in thường hết: nguyen
        if (strstr(strlwr(t->data.fullName), strlwr(name)) !=
            NULL)
        {
            // sau khi kiểm tra xong thì chuyển lại đúng chuẩn:
            formatString(t->data.fullName); // Hàm này trong file
            printf("| %-7d ", count++);
            outputARoomRenterByHorizontal(t->data);
        }
        else
        {
            // nếu không có thì
            formatString(t->data.fullName); // Hàm này trong file
        }
    }
    printf("+ ----- + ----- + ----- + ----- + \n");
}
```

3.5.8. In danh sách người thuê

- Duyệt qua danh sách người thuê.
- In ra người thuê theo chiều ngang đồng thời đưa tên người thuê theo đúng chuẩn.

```
void printRoomRenters(RoomRenters roomRenters)
{
    FILE *fileOut = fopen("../File/roomRenter/RoomRenters.txt",
"w");
    if (fileOut == NULL)
    {
        printf("\n\t%50c(*) In file khong thanh cong (*)\n\a", '
');
    }
    else
    {
        int count = 1;
        fprintf(fileOut, "%s", "+ ----- + -----
----- + ----- + ----- + -----
----- + ----- + \n");
        fprintf(fileOut, "%s", "|      STT      |              Ho va ten
|   Ngay sinh   | Gioi tinh | So dien thoai |              So CCCD
| So phong     | \n");
        fprintf(fileOut, "%s", "+ ----- + -----
----- + ----- + ----- + -----
----- + ----- + \n");
        for (Node *t = roomRenters.pHead; t != NULL; t = t-
>pNext)
        {
            fprintf(fileOut, "%c %-7d ", '|', count++);
            char sex[4];
            if (t->data.sex == 0)
                strcpy(sex, "Nu");
            else
                strcpy(sex, "Nam");
            fprintf(fileOut, "%c %-30s%c %-2d%c%-2d%c%-4d
%c %-9s %c %-15s %c %-20s %c %-10d %c\n", '|', t->data.fullName,
'|', t->data.birthDay.day, '/', t->data.birthDay.month, '/', t-
>data.birthDay.year, '|', sex, '|', t->data.phoneNumber, '|', t-
>data.identityCard, '|', t->data.room->number, '|');
        }
        fprintf(fileOut, "%s", "+ ----- + -----
----- + ----- + ----- + -----
----- + ----- + \n");
        printf("\n\t%50c(*) In file thanh cong (*)\n", ' ');
    }
    fclose(fileOut);
}
```

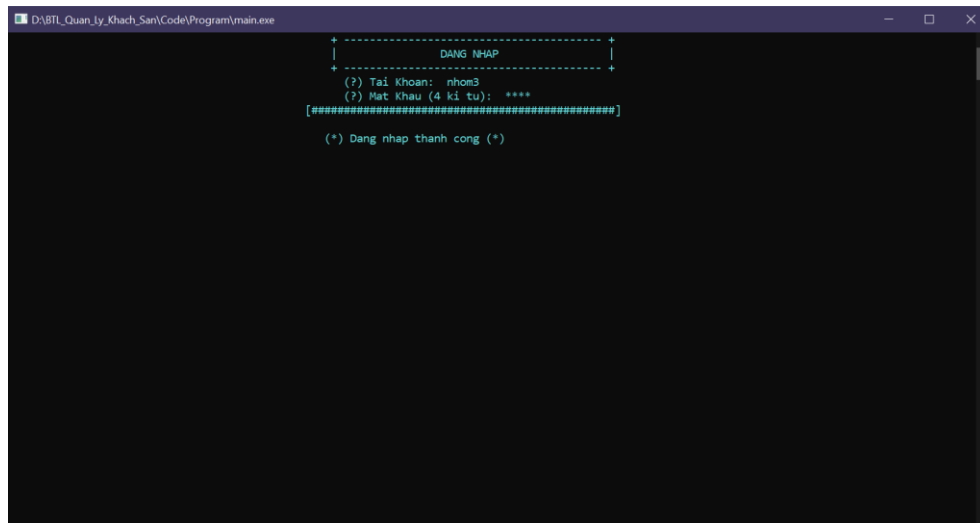

3.5.9. Lưu thay đổi

- Duyệt qua danh sách người thuê, danh sách các phòng
- Ghi vào file để lưu lại thay đổi

```
void saveFile(RoomRenters roomRenters, Room rooms[], int n)
{
    printf("\n\t%40c(*) Dang luu du lieu danh sach phong (*)\n",
    ' ');
    Sleep(1000); // delay 2s
    writeRooms(rooms, n);
    printf("\n\t%40c(*) Dang luu du lieu danh sach nguoi thue
    (*)\n", ' ');
    Sleep(1000); // delay 2s
    writeRoomRenters(roomRenters);
}
```

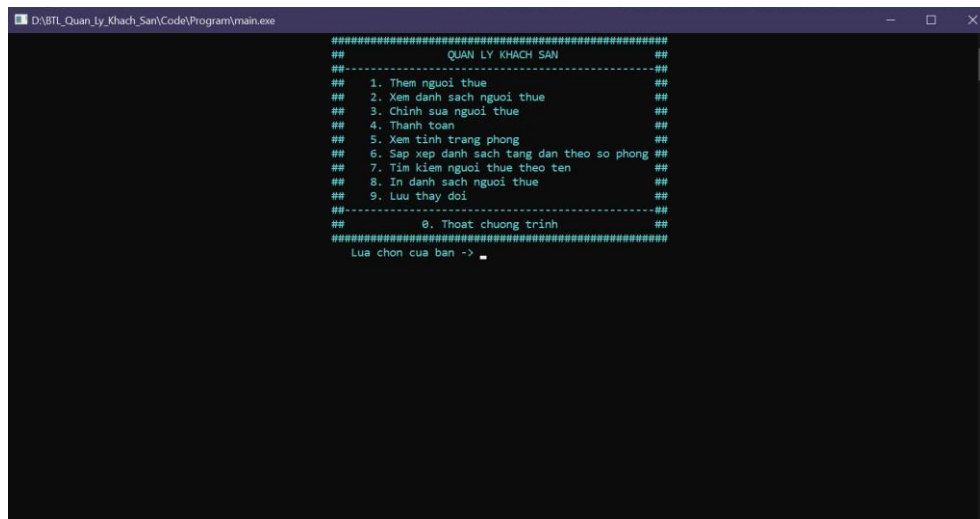
CHƯƠNG 4: THỬ NGHIỆM ỨNG DỤNG

4.1. Giao diện đăng nhập



Cũng như bao ứng dụng, chương trình khác. Đầu tiên chúng ta tiến hành đăng nhập để có quyền truy cập vào hệ thống. Tài khoản và mật khẩu đã được lưu trước đó, chỉ có quản trị viên mới biết và có quyền truy cập.

4.2. Giao diện sau khi đăng nhập



Đây là giao diện chính để chúng ta sử dụng. Có đầy đủ các chức năng hiệu quả để quản lý khách sạn.

4.3. Giao diện thêm người thuê

D:\BT1_Quan_ly_Khach_San\Code\Program\main.exe

THÔNG TIN NGƯỜI THUÊ VUA NHẬP

Họ và tên: Nguyen Van A
Ngày sinh: 23/11/1998
Giới tính: Nam
Số điện thoại: 0784566371
Số CCCD: 123321789456
Số phòng: 401

STT	Họ và tên	Ngày sinh	Giới tính	Số điện thoại	Số CCCD	Số phòng
1	Hoang Gia Kiet	23/03/2001	Nam	1111111111	111111111111	201
2	Nguyen Thi Thanh Nhu	19/09/2002	Nu	2222222222	22222222222	301
3	Le Trung Quyen	25/12/2003	Nam	3333333333	33333333333	302
4	Vo Thi Tuong Vi	01/04/2000	Nu	4444444444	44444444444	304
5	Dao Hoang Thu Kim	18/04/1998	Nu	5555555555	55555555555	402
6	Tran Nguyen Huu Hoang	24/06/2003	Nam	6666666666	66666666666	503
7	Lam Tam Nhu	05/06/2001	Nu	9999999999	99999999999	303
8	Nguyen Thanh Tung	05/07/1994	Nam	9998475473	777555666323	404
9	Cristiano Ronaldo	05/02/1985	Nam	1121121121	11211211212	203
10	Lionel Messi	24/06/1987	Nam	1221222211	12212121212	204
11	Ninh Duong Lan Ngoc	04/04/1990	Nu	8947482954	543454237623	502
12	Eimi Fukada	18/03/1998	Nu	4425412134	542123143542	501
13	Yuji Nishida	30/01/2000	Nam	3211233211	333123213213	403
14	Nguyen Van A	23/11/1998	Nam	0784566371	123321789456	401

Press any key to continue . . .

Đây là giao diện cho phép thêm một người thuê mới. Tất cả các thông tin người thuê mới phải được nhập vào.

4.4. Giao diện xem danh sách người thuê

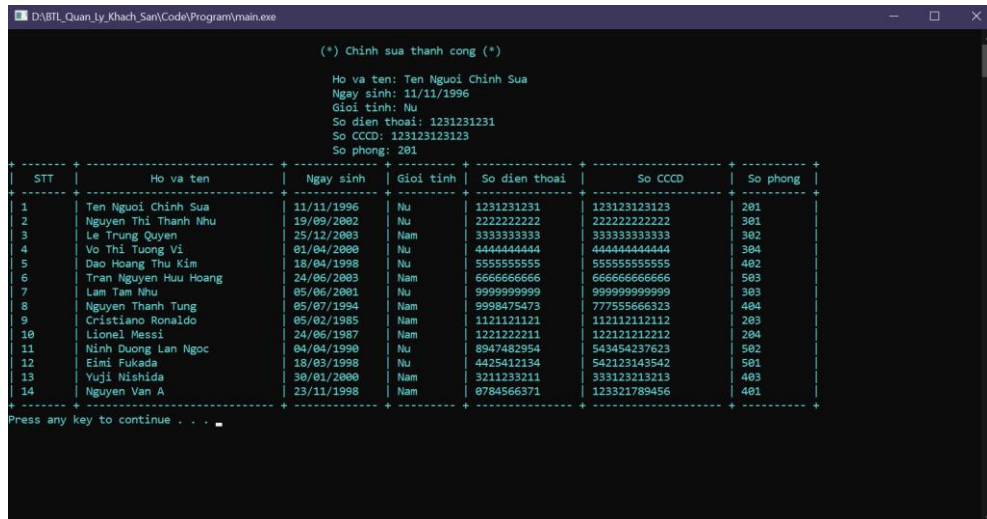
D:\BT1_Quan_ly_Khach_San\Code\Program\main.exe

2. XEM DANH SÁCH NGƯỜI THUÊ

STT	Họ và tên	Ngày sinh	Giới tính	Số điện thoại	Số CCCD	Số phòng
1	Hoang Gia Kiet	23/03/2001	Nam	1111111111	111111111111	201
2	Nguyen Thi Thanh Nhu	19/09/2002	Nu	2222222222	22222222222	301
3	Le Trung Quyen	25/12/2003	Nam	3333333333	33333333333	302
4	Vo Thi Tuong Vi	01/04/2000	Nu	4444444444	44444444444	304
5	Dao Hoang Thu Kim	18/04/1998	Nu	5555555555	55555555555	402
6	Tran Nguyen Huu Hoang	24/06/2003	Nam	6666666666	66666666666	503
7	Lam Tam Nhu	05/06/2001	Nu	9999999999	99999999999	303
8	Nguyen Thanh Tung	05/07/1994	Nam	9998475473	777555666323	404
9	Cristiano Ronaldo	05/02/1985	Nam	1121121121	11211211212	203
10	Lionel Messi	24/06/1987	Nam	1221222211	12212121212	204
11	Ninh Duong Lan Ngoc	04/04/1990	Nu	8947482954	543454237623	502
12	Eimi Fukada	18/03/1998	Nu	4425412134	542123143542	501
13	Yuji Nishida	30/01/2000	Nam	3211233211	333123213213	403
14	Nguyen Van A	23/11/1998	Nam	0784566371	123321789456	401

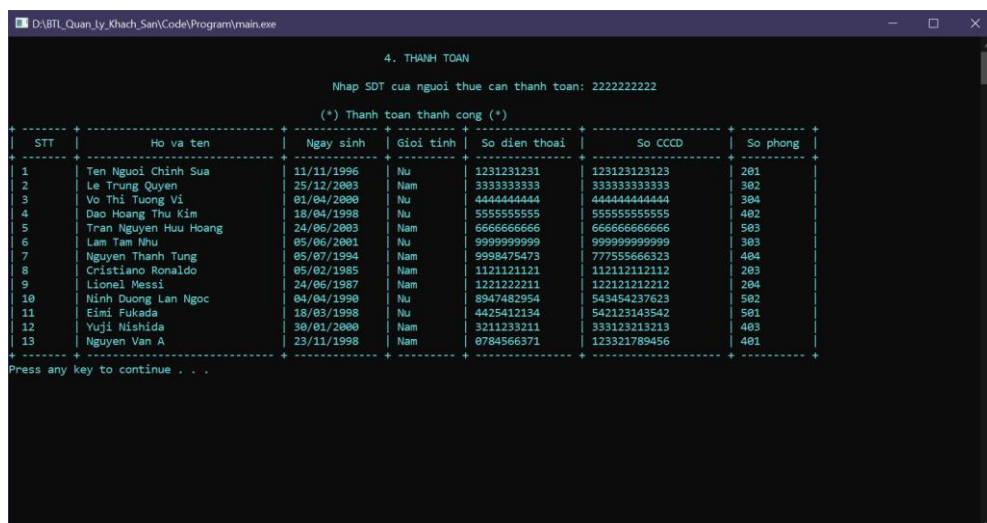
Press any key to continue . . .

4.5. Giao diện chỉnh sửa người thuê



Đây là giao diện khi cần chỉnh sửa một người thuê. Người thuê có thể lựa chọn tiêu chí cần chỉnh sửa.

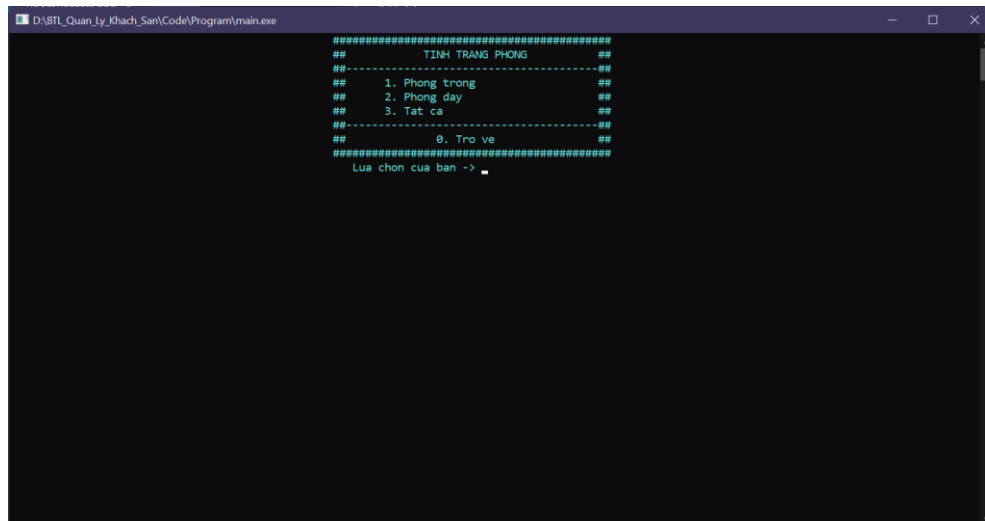
4.6. Giao diện thanh toán



Giao diện này sẽ xuất ra thông tin người thuê sau khi đã thanh toán xong. Sau khi thanh toán người thuê này sẽ bị xóa khỏi danh sách.

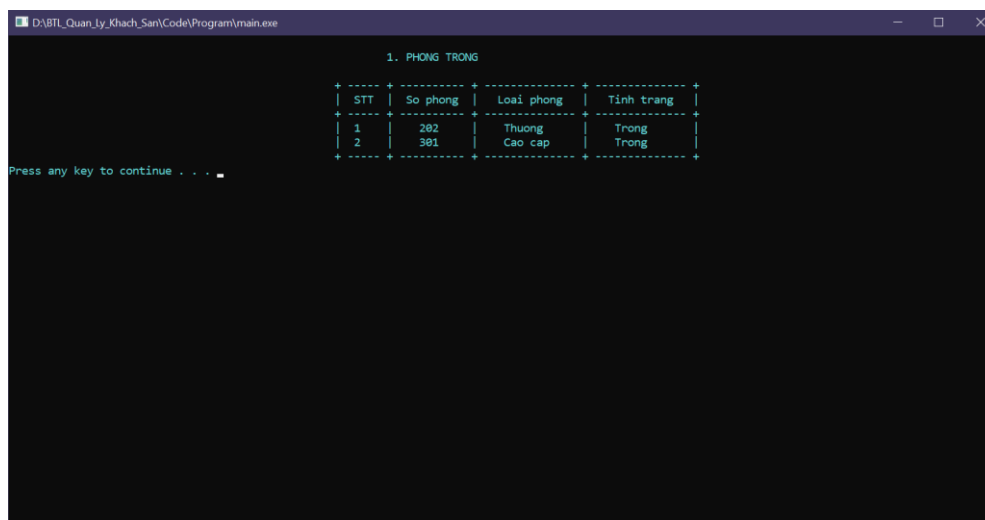
4.7. Giao diện xem tình trạng phòng

4.7.1. Giao diện menu sau khi chọn chức năng xem tình trạng phòng

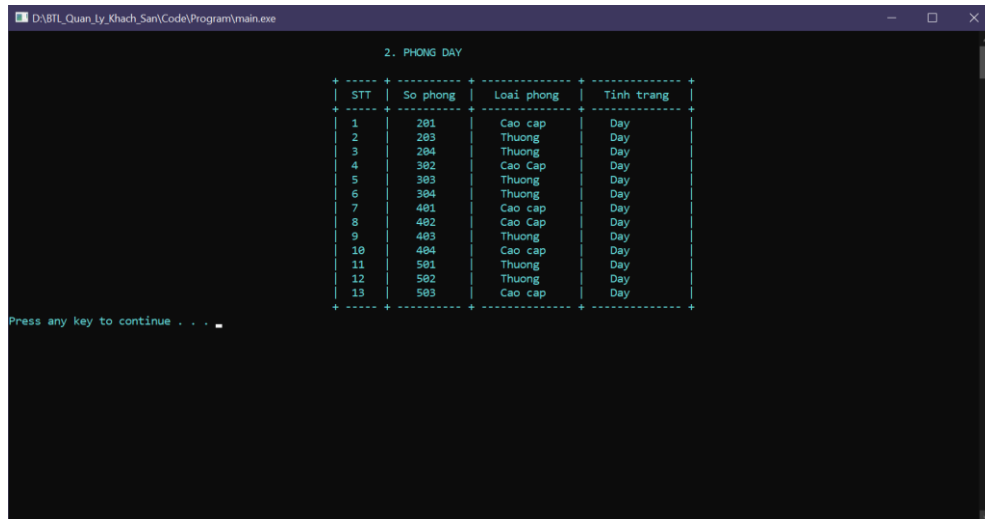


- Giao diện này cho phép chúng ta nhập lựa chọn muốn xem tình trạng phòng đầy, trống hay tắt cả các phòng.

4.7.2. Giao diện phòng trống

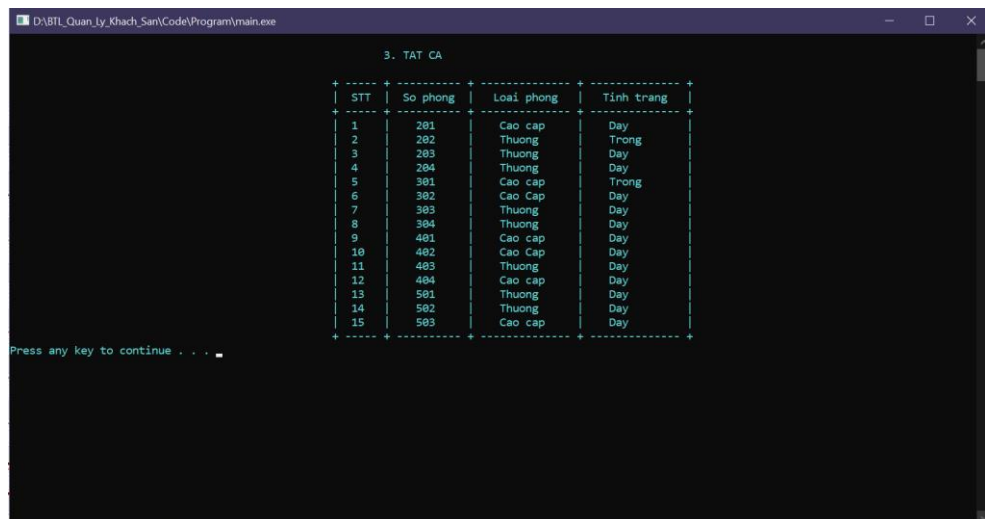


4.7.3. Giao diện phòng đầy



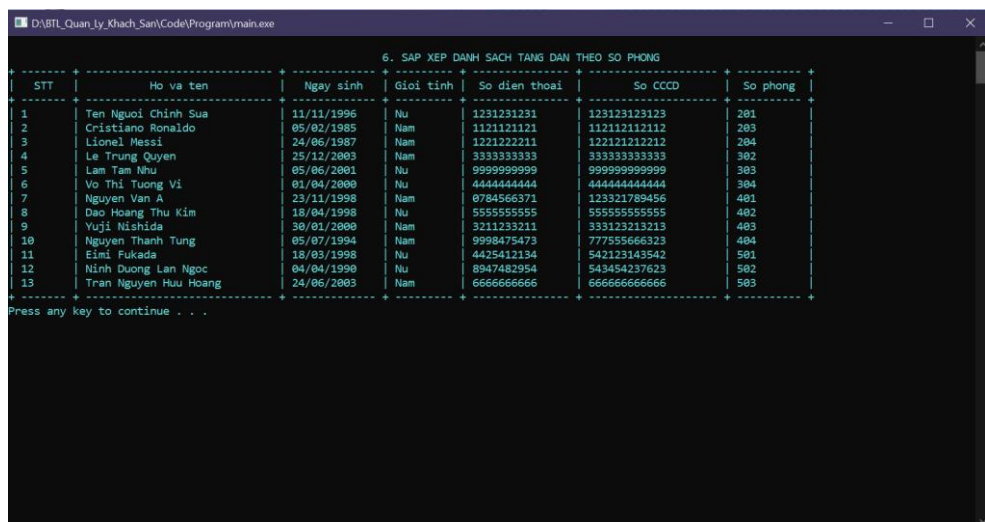
STT	So phong	Loại phòng	Tình trạng
1	201	Cao cap	Day
2	203	Thuong	Day
3	204	Thuong	Day
4	302	Cao Cap	Day
5	303	Thuong	Day
6	304	Thuong	Day
7	401	Cao cap	Day
8	402	Cao Cap	Day
9	403	Thuong	Day
10	404	Cao cap	Day
11	501	Thuong	Day
12	502	Thuong	Day
13	503	Cao cap	Day

4.7.4. Giao diện tình trạng tất cả các phòng



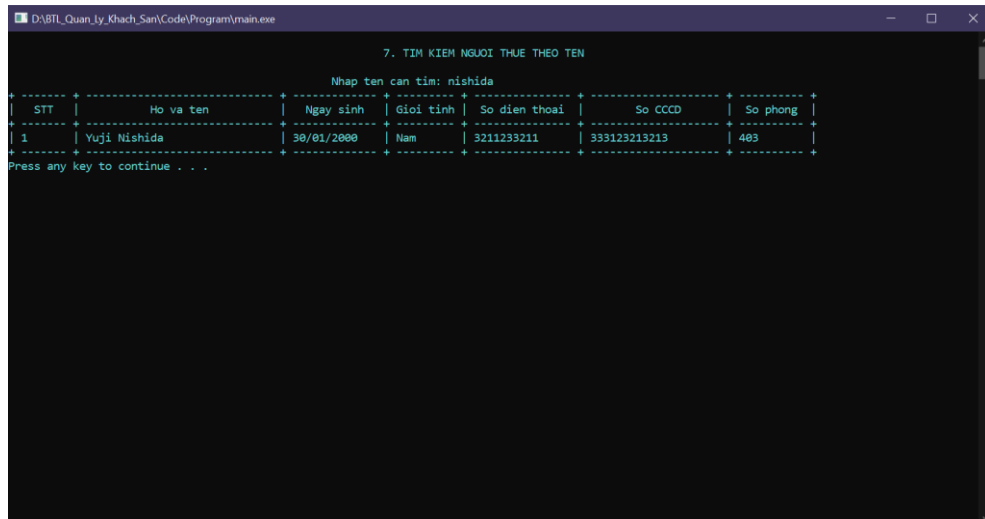
STT	So phong	Loại phòng	Tình trạng
1	201	Cao cap	Day
2	202	Thuong	Trong
3	203	Thuong	Day
4	204	Thuong	Day
5	301	Cao cap	Trong
6	302	Cao Cap	Day
7	303	Thuong	Day
8	304	Thuong	Day
9	401	Cao cap	Day
10	402	Cao Cap	Day
11	403	Thuong	Day
12	404	Cao cap	Day
13	501	Thuong	Day
14	502	Thuong	Day
15	503	Cao cap	Day

4.8. Giao diện sắp xếp người thuê theo số phòng



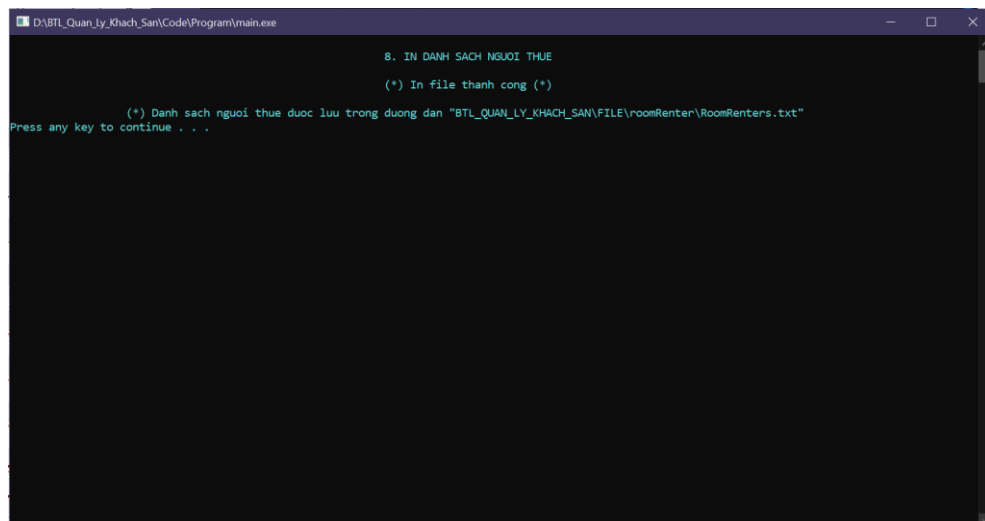
STT	Ho va ten	Ngay sinh	Gioi tinh	So dien thoai	So CCCD	So phong
1	Ten Ngươi Chính Sua	11/11/1996	Nu	1231231231	123123123123	201
2	Cristiano Ronaldo	05/02/1985	Nam	1121121121	112112112112	203
3	Lionel Messi	24/06/1987	Nam	1221222211	122121212212	204
4	Le Trung Quyên	25/12/2003	Nam	3333333333	333333333333	302
5	Lam Tam Nhu	05/06/2001	Nu	9999999999	999999999999	303
6	Vo Thi Tuong Vi	01/04/2000	Nu	4444444444	444444444444	304
7	Nguyen Van A	23/11/1998	Nam	0784566371	123321789456	401
8	Dao Hoang Thu Kim	18/04/1998	Nu	5555555555	555555555555	402
9	Yuji Nishida	30/01/2000	Nam	3211233211	333123213213	403
10	Nguyen Thanh Tung	05/07/1994	Nam	9998475473	777555666323	404
11	Eimi Fukada	18/03/1998	Nu	4425412134	542123143542	501
12	Ninh Duong Lan Ngoc	04/04/1990	Nu	8047482954	543454237623	502
13	Tran Nguyen Huu Hoang	24/06/2003	Nam	6666666666	666666666666	503

4.9. Giao diện tìm kiếm người thuê theo tên



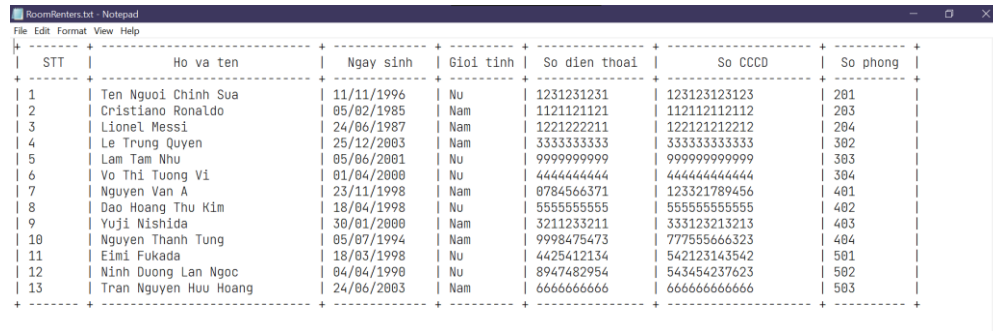
Giao diện này cho phép nhập tên người thuê (Nhập đúng chuẩn hay không đúng chuẩn) cần tìm kiếm, chương trình xuất ra tên người thuê cần tìm kiếm.

4.10. Giao diện in danh sách người thuê



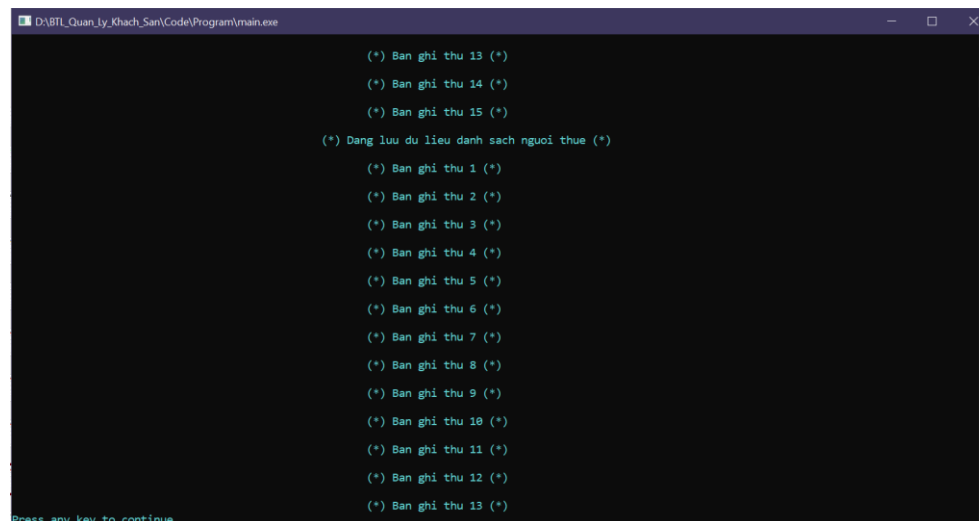
Ở giao diện này, danh sách người thuê được in ra file RoomRenters.txt

4.10.1. File sau khi in

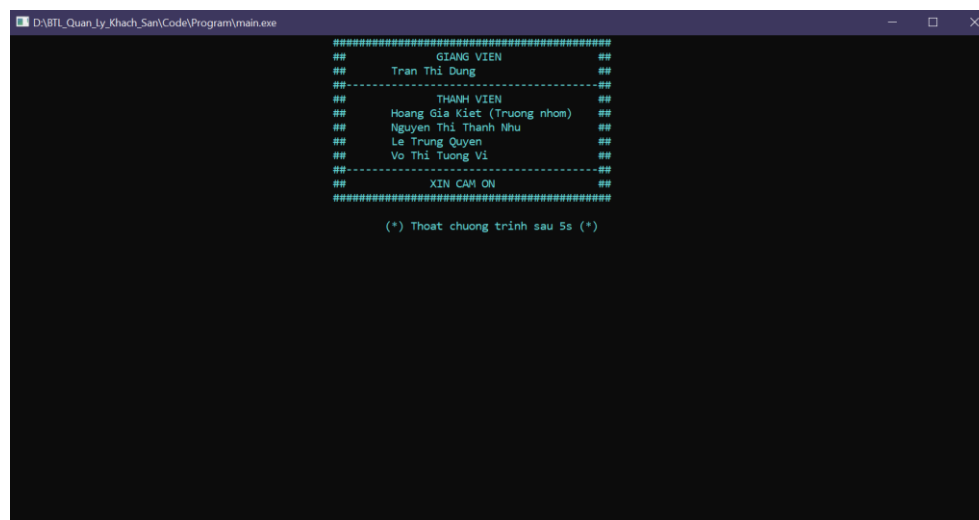


STT	Ho va ten	Ngay sinh	Gioi tinh	So dien thoai	So CCCD	So phong
1	Ten Ngươi Chính Sua	11/11/1996	Nu	1231231231	123123123123	201
2	Cristiano Ronaldo	05/02/1985	Nam	1121121212	112112112112	203
3	Lionel Messi	24/06/1987	Nam	1221222211	122121212212	204
4	Le Trung Quyen	25/12/2003	Nam	3333333333	333333333333	302
5	Lam Tam Nhu	05/06/2001	Nu	9999999999	999999999999	303
6	Vo Thi Tuong Vi	01/04/2000	Nu	4444444444	444444444444	304
7	Nguyen Van A	23/11/1998	Nam	0784566371	123321789456	401
8	Dao Hoang Thu Kim	18/04/1998	Nu	5555555555	555555555555	402
9	Vuji Nishida	30/01/2000	Nam	3211233211	333123213213	403
10	Nguyen Thanh Tung	05/07/1994	Nam	9998475473	777555666323	404
11	Eimi Fukada	18/03/1998	Nu	4425412134	542123143542	501
12	Ninh Duong Lan Ngoc	04/04/1990	Nu	8947482954	543454237623	502
13	Tran Nguyen Huu Hoang	24/06/2003	Nam	6666666666	666666666666	503

4.11. Giao diện lưu thay đổi



4.12. Giao diện sau khi thoát



CHƯƠNG 5 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Kết quả đạt được

Chương trình đã được phát triển thành công. Đảm bảo đầy đủ các tính năng cơ bản để quản lý khách sạn một cách hiệu quả, thuận lợi.

Chương trình hoạt động tốt trên máy tính, laptop. Hiệu suất nhanh chóng ổn định, giao diện thân thiện, dễ sử dụng, dễ thao tác.

5.1.2. Kiến nghị

Do thời gian tìm hiểu, phân tích, thiết kế chương trình còn hạn hẹp nên chương trình vẫn còn một số hạn chế: Chưa có tính năng in hóa đơn, thành tiền cho người thuê sau khi thanh toán.

5.2. Hướng phát triển

Xây dựng hoàn thiện chương trình, thêm các tính năng như in hóa đơn thanh toán,...

Tìm hiểu kết hợp nhiều ngôn ngữ khác để phát triển chương trình thành phần mềm, ứng dụng.

TÀI LIỆU THAM KHẢO

- [1]. <https://teky.edu.vn/blog/github-la-gi/>
- [2]. https://freetuts.net/lich-su-va-dac-diem-cua-ngon-ngu-lap-trinh-c-2077.html?fbclid=IwAR1CodWLLjV4f0c2R-_9pu0xKdZmJk5G4UypAshb6_MuECCChzcqNk3IhM
- [3]. Giáo trình tin học đại cương PGS.TS Phạm Văn Ất
<https://thuvien.utc2.edu.vn/hoclieu/viewbook/?id={5766516F-29C9-44EA-AE59-904A9203AE00}>