|  |  |  |
| --- | --- | --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC NHA TRANG**  **KHOA CÔNG NGHỆ THÔNG TIN** | | |
| Trang chủ trường Đại học Nha Trang - Nha Trang University | | |
| **BÁO CÁO THỰC TẬP CƠ SỞ**  **QUẢN LÝ VIỆC SỬ DỤNG ĐIỆN CỦA KHÁCH HÀNG** | | |
|  | **Giảng viên hướng dẫn** | **: ThS. Huỳnh Tuấn Anh** |
|  | **Sinh viên thực hiện** | **: Ngô Thục Trí Đạt** |
|  | **Mã số sinh viên** | **: 6113478** |
| Khánh Hòa – 2021 | | |

|  |  |  |
| --- | --- | --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC NHA TRANG**  **KHOA CÔNG NGHỆ THÔNG TIN** | | |
| Trang chủ trường Đại học Nha Trang - Nha Trang University | | |
| **BÁO CÁO THỰC TẬP CƠ SỞ**  **QUẢN LÝ VIỆC SỬ DỤNG ĐIỆN CỦA KHÁCH HÀNG** | | |
|  | GVHD | : ThS. Huỳnh Tuấn Anh |
|  | SVTH | : Ngô Thục Trí Đạt |
|  | MSSV | : 6113478 |
| Khánh Hòa – Tháng 12/2021 | | |

**MỤC LỤC**

**Trang**

[**I. Các khái niệm** 9](#_Toc93733479)

[1.1. Ngôn ngữ lập trình C++ 9](#_Toc93733480)

[1.2. Dev-C++ 10](#_Toc93733481)

[**II. Tổng quan** 10](#_Toc93733482)

[2.1. Tổ chức dữ liệu 10](#_Toc93733483)

[2.2. Giới thiệu về chương trình 10](#_Toc93733484)

[2.3. Nhược điểm 11](#_Toc93733485)

[**III. Một số phần quan trọng trong chương trình** 11](#_Toc93733486)

[3.1. Struct (cấu trúc) 11](#_Toc93733487)

[3.2. Switch case 11](#_Toc93733488)

[3.3. pop\_back() và push\_back() 11](#_Toc93733489)

[3.4. Các lệnh được sử dụng 11](#_Toc93733490)

[3.5. Terminal 12](#_Toc93733491)

[**IV. Chương trình** 12](#_Toc93733492)

[4.1. Giao diện chính 12](#_Toc93733493)

[4.2. Chương trình tính tiền điện 13](#_Toc93733494)

[4.3. Các chức năng 13](#_Toc93733495)

[4.3.1. Thêm thông tin khách hàng (Case 1) 13](#_Toc93733496)

[4.3.2. Xóa thông tin khách hàng (Case 2) 15](#_Toc93733497)

[4.3.3. Cập nhật khách hàng (Case 3) 17](#_Toc93733498)

[4.3.4. Thêm hóa đơn (Case 4) 19](#_Toc93733499)

[4.3.5. Xóa hóa đơn (Case 5) 21](#_Toc93733500)

[4.3.6. Tìm kiếm khách hàng (Case 6) 23](#_Toc93733501)

[4.3.7. Tìm kiếm hóa đơn (Case 7) 24](#_Toc93733502)

[4.3.8. In danh sách khách hàng và danh sách hóa đơn (Case 8 + 9) 25](#_Toc93733503)

[4.3.9. anyKey() 26](#_Toc93733504)

**Danh Mục Hình Ảnh**

[Hình 1. Tổ chức dữ liệu 10](#_Toc93733505)

[Hình 2. Dữ liệu dạng nhị phân 10](#_Toc93733506)

[Hình 3. Giao diện terminal 12](#_Toc93733507)

[Hình 4. Code chương trình chính 13](#_Toc93733508)

[Hình 5. Chương trình tính tiền điện 13](#_Toc93733509)

[Hình 6. Code thêm thông tin khách hàng 14](#_Toc93733510)

[Hình 7. Nhập thông tin khách hàng 14](#_Toc93733511)

[Hình 8.Thêm khách hàng thành công và in ra danh sách khách hàng 15](#_Toc93733512)

[Hình 9. Code xóa thông tin khách hàng 15](#_Toc93733513)

[Hình 10.Hàm xóa 16](#_Toc93733514)

[Hình 11. Danh sách khách hàng trước xóa 16](#_Toc93733515)

[Hình 12.Danh sách khách hàng sau xóa 17](#_Toc93733516)

[Hình 13. Cập nhật thông tin khách hàng 17](#_Toc93733517)

[Hình 14.Hàm cập nhật 18](#_Toc93733518)

[Hình 15.Danh sách trước cập nhật 18](#_Toc93733519)

[Hình 16. Danh sách sau cập nhật 19](#_Toc93733520)

[Hình 17.Thông tin thay đổi 19](#_Toc93733521)

[Hình 18.Code thêm thông tin hóa đơn 20](#_Toc93733522)

[Hình 19. Hàm nhập thông tin hóa đơn 20](#_Toc93733523)

[Hình 20. Hàm thêm hóa đơn 20](#_Toc93733524)

[Hình 21. Nhập thông tin hóa đơn và danh sách sau thêm 21](#_Toc93733525)

[Hình 22. Xóa hóa đơn 21](#_Toc93733526)

[Hình 23. Hàm xóa hóa đơn 22](#_Toc93733527)

[Hình 24. Danh sách hóa đơn trước xóa 22](#_Toc93733528)

[Hình 25. Danh sách hóa đơn sau xóa 23](#_Toc93733529)

[Hình 26. Tìm kiếm khách hàng 23](#_Toc93733530)

[Hình 27.Hàm tìm kiếm khách hàng 24](#_Toc93733531)

[Hình 28. Tìm kiếm khách hàng 001 24](#_Toc93733532)

[Hình 29.Tìm kiếm hóa đơn 24](#_Toc93733533)

[Hình 30. Hàm tìm kiếm hóa đơn 25](#_Toc93733534)

[Hình 31. Tìm kiếm hóa đơn khách 45 25](#_Toc93733535)

[Hình 32. In danh sách khách hàng và danh sách hóa đơn 25](#_Toc93733536)

[Hình 33. Hàm anyKey() 26](#_Toc93733537)

**MỞ ĐẦU**

**Lý do chọn đề tài**

Trong thời đại mà tự động hoá và trí tuệ nhân tạo ngày một phổ biến rộng rãi, nhằm đơn giản hóa việc quản lí và các hoạt động liên quan, hầu hết các công việc đều cần đến sự trợ giúp trợ giúp của máy tính. Các loại hóa đơn dịch vụ cũng sử dụng máy tính thay thế cho việc ghi chép thủ công bằng giấy mực bởi tiềm năng của nó.  
 Việc ghi chép thủ công thông qua sổ sách thường xuất hiện các lỗi gây nhầm lẫn, không đáng có. Việc quản lí thông tin khách hàng và hóa đơn của họ nhằm tránh sai sót, nhầm lẫn gây thiệt hại về quyền lợi của khách hàng.  
 Ngày nay, với sự trợ giúp của công nghệ máy tính, có khả năng lưu trữ toàn bộ thông tin của khách hàng như thông tin cá nhân hay hóa đơn hàng tháng. Với giao diện dễ tiếp cận, nhiều chức năng được tối ưu, giúp hệ thống dễ dàng kiểm soát, đảm bảo quyền lợi cho khách hàng, cũng như chủ động liên hệ với họ khi cần thiết.  
 Từ những lý do trên, em quyết định chọn đề tài: “**Quản lý việc sử dụng điện của khách hàng**”.

* + **Mục tiêu nghiên cứu:**  
    - ***Mục tiêu chung:*** Xây dựng chương trình quản lí việc sử dụng điện của khách hàng **-** ***Mục tiêu cụ thể:***   
    + Tổ chức lưu trữ dữ liệu **+** Tạo hóa đơn sử dụng điện  
    + Tạo hàm tính tiền điện theo bậc thang  
    + Tạo các chức năng cơ bản cho khách hàng**+** Tạo các chức năng bổ sung như tìm kiếm, sắp xếp
  + **Nội dung và phương pháp thực hiện:**

**- *Nội dung thực hiện:***+ Tìm hiểu các kiến thức, yêu cầu đối với việc quản lí sử dụng điện của khách hàng   
+ Thu thập dữ liệu về giá tiền điện bậc thang  
+ Xây dựng chương trình  
- ***Phương pháp:***

**+** Chọn lọc các tiêu chí, yêu cầu  
+ Tạo hóa đơn điện  
+ Xây dựng chương trình với các chức năng cần thiết bằng ngôn ngữ C++

**+** Sử dụng cấu trúc để lưu thông tin

**+** Lưu trữ dữ liệu dưới dạng tệp nhị phân

# **I. Các khái niệm**

## 1.1. Ngôn ngữ lập trình C++

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

C++ được thiết kế hướng tới lập trình hệ thống máy tính và phần mềm nhúng trên các mạch vi xử lý, bao gồm cả hệ thống có tài nguyên hạn chế và tài nguyên khổng lồ, với ưu điểm vượt trội về hiệu suất, hiệu quả và tính linh hoạt cao. C ++ có thể tìm thấy ở mọi nơi, với những điểm mạnh là cơ sở hạ tầng phần mềm và các ứng dụng bị hạn chế tài nguyên. bao gồm: phần mềm ứng dụng , trò chơi điện tử, các hệ thống máy chủ và các ứng dụng ưu tiên về hiệu suất . C++ có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, Linux, các phiên bản Unix. Nhiều nhà cung cấp cung cấp các trình biên dịch C ++, bao gồm Tổ chức Phần mềm Tự do, Microsoft, Intel và IBM.

C++ được Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) chuẩn hóa, với phiên bản tiêu chuẩn mới nhất được ISO phê chuẩn và công bố vào tháng 12 năm 2017 là ISO / IEC 14882: 2017 (được gọi một cách không chính thức là C++ 17). Trước khi tiêu chuẩn hóa ban đầu vào năm 1998, C++ được phát triển bởi Bjarne Stroustrup tại Bell Labs từ năm 1979, như một phần mở rộng của ngôn ngữ C khi ông muốn một ngôn ngữ hiệu quả và linh hoạt tương tự như C, cũng cung cấp các tính năng cấp cao cho tổ chức chương trình. C++ 20 là tiêu chuẩn được lên kế hoạch tiếp theo sau đó, phù hợp với chuỗi hiện tại của một phiên bản mới cứ sau ba năm.

Nhiều ngôn ngữ lập trình khác được phát triển dựa trên nền tảng C++, bao gồm C#, D, Java và các phiên bản mới hơn của C.

## 1.2. Dev-C++

Dev-C++ là một môi trường phát triển tích hợp tự do (IDE) được phân phối dưới hình thức giấy phép Công cộng GNU hỗ trợ việc lập trình bằng C/C++. Nó cũng nằm trong bộ trình dịch mã nguồn mở MinGW. Chương trình IDE này được viết bằng ngôn ngữ Delphi.

Dự án phát triển Dev-C++ được lưu trữ trên SourceForge. Dev-C++ nguyên được phát triển bởi một lập trình viên có tên là Colin Laplace và chỉ chạy trên hệ điều hành Microsoft Windows.

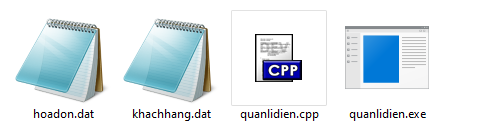
# **II. Tổng quan**

## 2.1. Tổ chức dữ liệu

- Chương trình quản lí được tổ chức như sau:

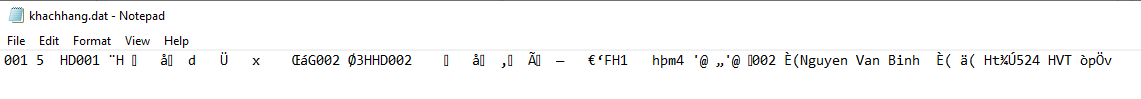
+ 2 file .dat lưu trữ dữ liệu

+ file nguồn .cpp



Hình 1. Tổ chức dữ liệu

+ dữ liệu lưu trữ dưới dạng nhị phân



Hình 2. Dữ liệu dạng nhị phân

## 2.2. Giới thiệu về chương trình

Chương trình quản lý sử dụng điện của khách hàng, bao gồm các chức năng cơ bản liên quan đến quản lý. Đối với khách hàng, có thêm xóa và cập nhật thông tin cho khách hàng. Đối với hóa đơn có nhập hóa đơn, tiền sẽ được tự động tính sử dụng hàm tính tiền điện. Có giao diện đơn giản, thuận tiện, dễ sử dụng

## 2.3. Nhược điểm

Vì chương trình chỉ sử dụng màn hình terminal để chạy và xuất các kết quả, nên không có giao diện đẹp. Các chức năng còn hạn chế, chưa đủ đáp ứng cho hệ thống lớn.

# **III. Một số phần quan trọng trong chương trình**

## 3.1. Struct (cấu trúc)

Dùng như một khung để nhập và lưu dữ liệu cho các loại đối tượng. Những dòng dữ liệu nhập vào lưu dưới dạng vector

## 3.2. Switch case

**Lệnh switch case** là một cấu trúc điều khiển & rẽ nhánh hoàn toàn có thể được thay thế bằng cấu trúc if else. Tuy nhiên, việc sử dụng switch case sẽ giúp code của chúng ta dễ viết và dễ đọc hơn. Thuận tiện trong việc điều khiển các chức năng

## 3.3. pop\_back() và push\_back()

**push\_back()**: Hàm đẩy một phần tử vào vị trí sau cùng của vector. Nếu kiểu của đối tượng được truyền dưới dạng tham số trong push\_back() không giống với kiểu của vector thì sẽ bị ném ra.

**pop\_back()**: Hàm pop\_back () được sử dụng để xóa đi phần tử cuối cùng một vector.

## 3.4. Các lệnh được sử dụng

- fflush(stdin): dùng để xóa bộ đệm

+ stdin là dòng vào tiêu chuẩn(bàn phím). Các hàm scanf, gets, getchar ... đều nhận dữ liệu từ stdin. Nếu trên stdin có đủ dữ liệu thì các hàm trên sẽ chỉ nhận một phần dữ liệu mà chúng yêu cầu. Phần dữ liệu còn lại(chưa được nhận) vẫn ở lại trên dòng nhập tiêu chuẩn(stdin). Khi trên stdin không đủ dữ liệu theo yêu cầu của các hàm, thì máy sẽ tàm dừng để người sử dụng nhập thêm dữ liệu từ bàn phím lên stdin cho đến khi bấm Enter.

- fopen() : mở file

- str.erase() : loại bỏ một phần của chuỗi hoặc vector

- strcmp(): So sánh chuỗi

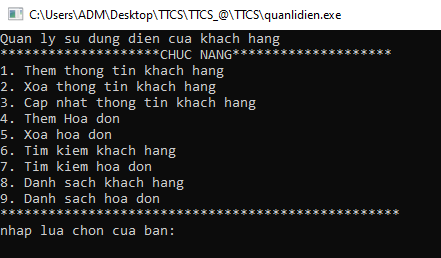
## 3.5. Terminal

Ứng dụng **Terminal** cho phép bạn điều khiển máy tính của mình bằng dấu nhắc lệnh (command prompt). Terminal là giao diện dòng lệnh (Command line interface)

# **IV. Chương trình**

## 4.1. Giao diện chính

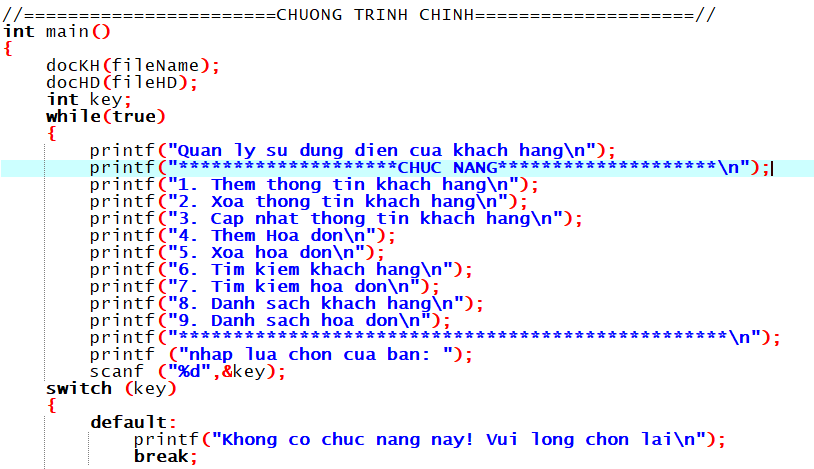
Terminal:



Hình 3. Giao diện terminal

Xây dựng khung sử dụng switch case. Nhập các số từ 1 – 9 sẽ tương ứng với các chức năng được xây dựng.

Code:

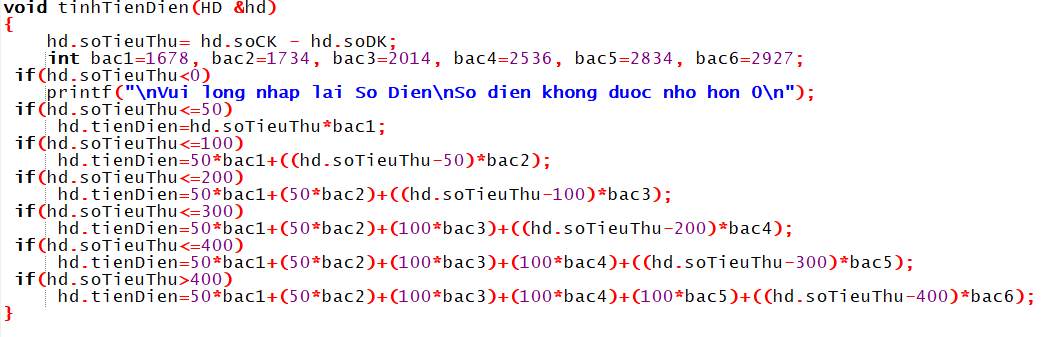


Hình 4. Code chương trình chính

## 4.2. Chương trình tính tiền điện

- Tiền điện sử dụng số điện tiêu thụ : Số điện tiêu thụ = Số cuối kì – số đầu kì

Code:



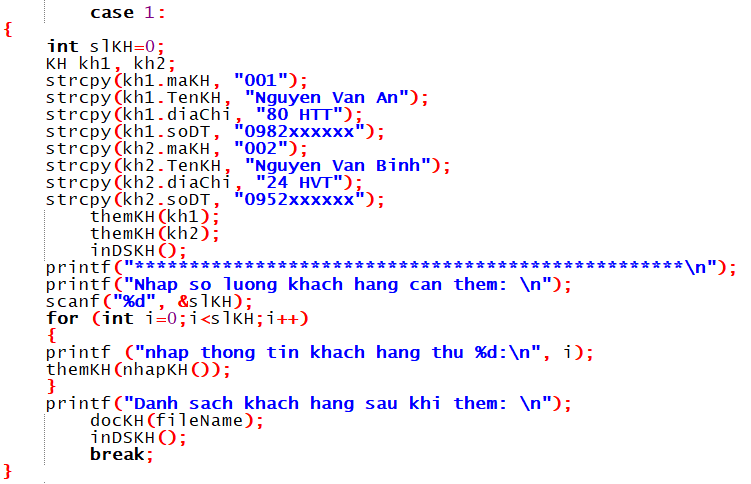
Hình 5. Chương trình tính tiền điện

## 4.3. Các chức năng

### 4.3.1. Thêm thông tin khách hàng (Case 1)

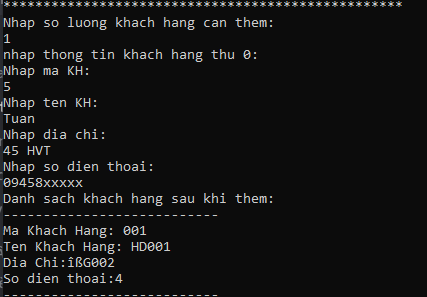
Thêm các thông tin của khách hàng theo cấu trúc: mã khách hàng, tên khách hàng, địa chỉ, số điện thoại

Code:

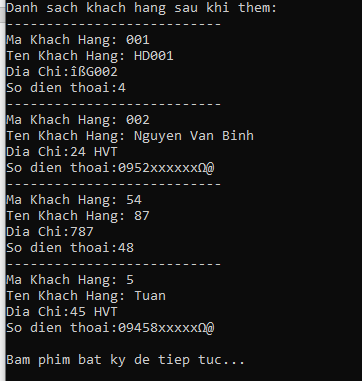


Hình 6. Code thêm thông tin khách hàng

Terminal:



Hình 7. Nhập thông tin khách hàng

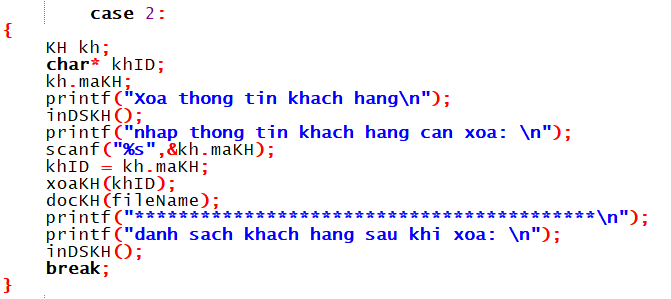


Hình 8.Thêm khách hàng thành công và in ra danh sách khách hàng

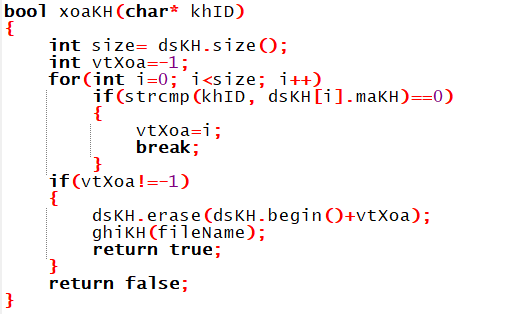
### 4.3.2. Xóa thông tin khách hàng (Case 2)

Xóa thông tin khách hàng bằng cách so sánh chuỗi, nhập mã khách hàng bất kì, nếu so sánh trùng khớp sẽ xóa thông tin của khách hàng đó.

Code:



Hình 9. Code xóa thông tin khách hàng

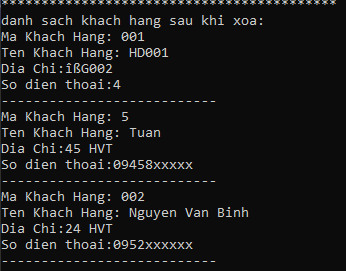


Hình 10.Hàm xóa

Terminal:



Hình 11. Danh sách khách hàng trước xóa

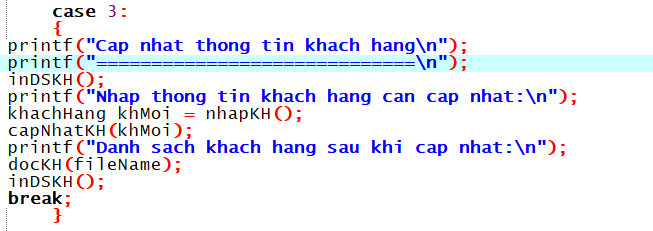


Hình 12.Danh sách khách hàng sau xóa

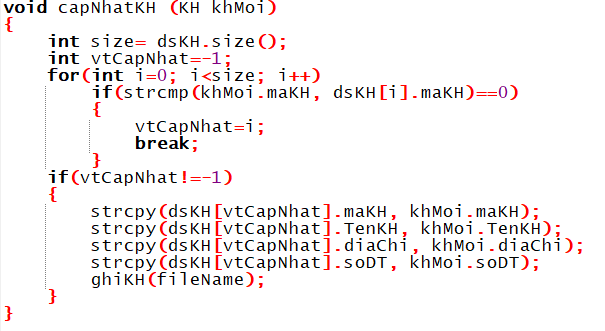
### 4.3.3. Cập nhật khách hàng (Case 3)

Nhập mã khách hàng, nếu mã trùng khớp với dữ liệu được ghi thì các dữ liệu nhập tiếp theo sẽ có hiệu lực thay đổi

Code:

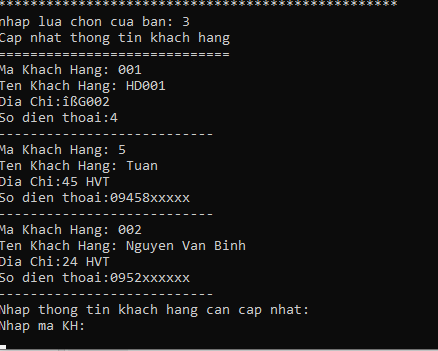


Hình 13. Cập nhật thông tin khách hàng

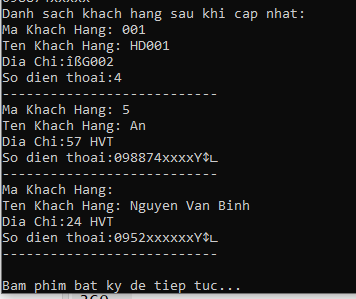


Hình 14.Hàm cập nhật

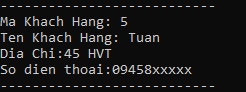
Terminal:

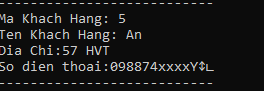


Hình 15.Danh sách trước cập nhật



Hình 16. Danh sách sau cập nhật



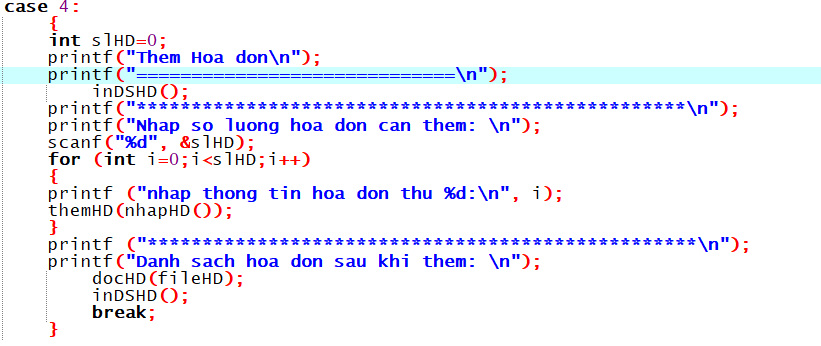


Hình 17.Thông tin thay đổi

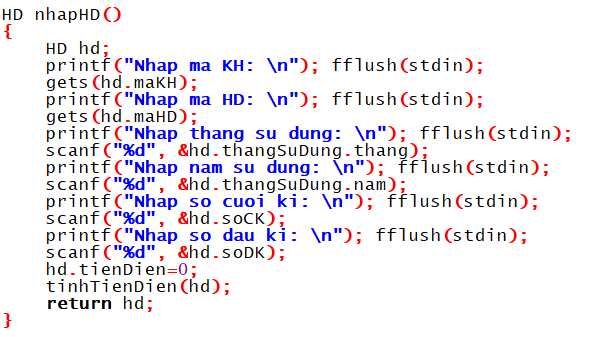
### 4.3.4. Thêm hóa đơn (Case 4)

Nhập số lượng hóa đơn muốn thêm mong muốn, sau đó nhập dữ liệu theo cấu trúc: mã khách hàng; mã hóa đơn; tháng, năm sử dụng; chỉ số đầu, cuối kì. Sau khi nhấn Enter thì số điện tiêu thụ và tiền điện sẽ được tính theo chương trình tính tiền điện.

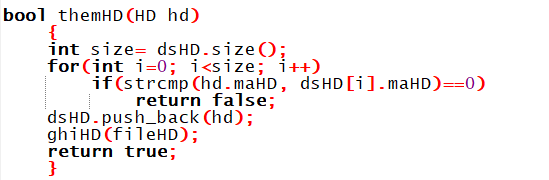
Code:



Hình 18.Code thêm thông tin hóa đơn

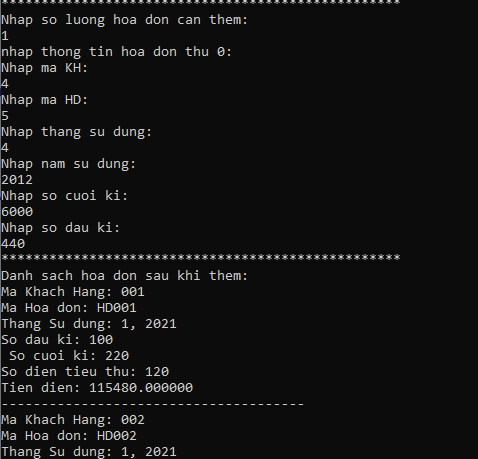


Hình 19. Hàm nhập thông tin hóa đơn



Hình 20. Hàm thêm hóa đơn

Terminal:

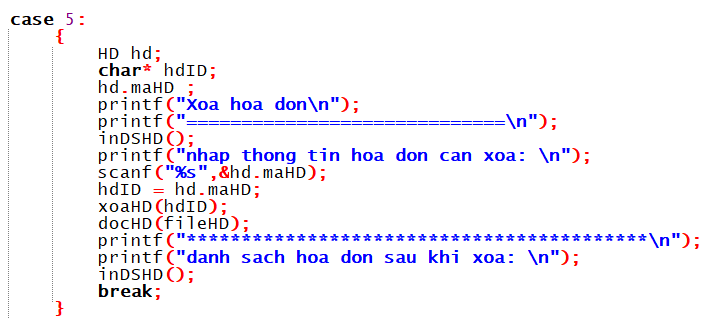


Hình 21. Nhập thông tin hóa đơn và danh sách sau thêm

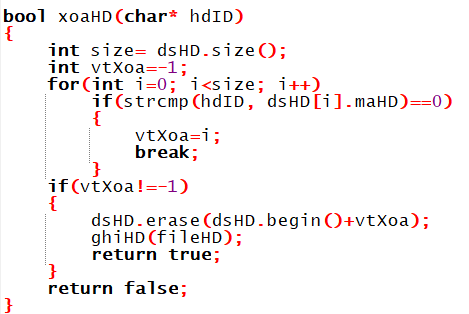
### 4.3.5. Xóa hóa đơn (Case 5)

Nhập chuỗi sau đó so sánh, nếu chuỗi khớp với mã hóa đơn thì thực hiện xóa

Code:



Hình 22. Xóa hóa đơn

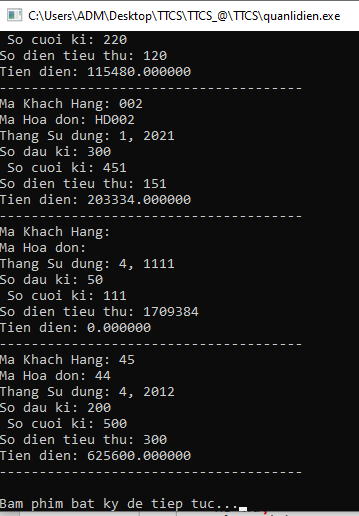


Hình 23. Hàm xóa hóa đơn

Terminal:



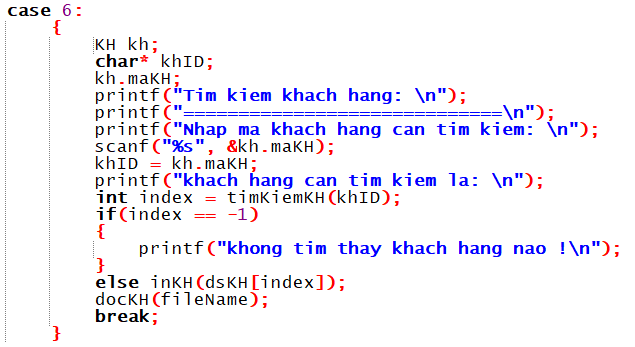
Hình 24. Danh sách hóa đơn trước xóa



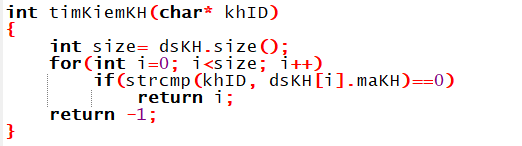
Hình 25. Danh sách hóa đơn sau xóa

### 4.3.6. Tìm kiếm khách hàng (Case 6)

So sánh chuỗi nhập vào, nếu trùng khớp với mã khách hàng thì in ra khách hàng. Nếu không trùng thì thông báo không tìm thấy khách hàng

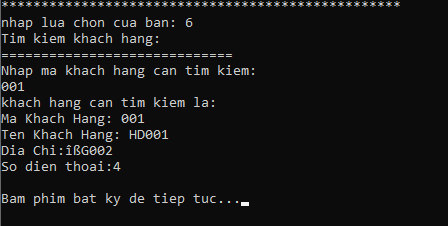
Code: 

Hình 26. Tìm kiếm khách hàng



Hình 27.Hàm tìm kiếm khách hàng

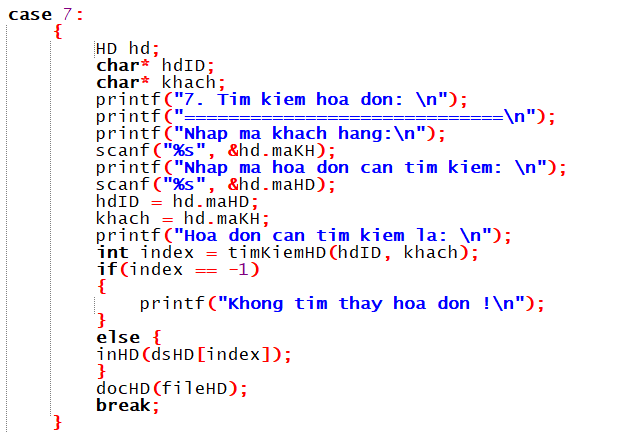
Terminal:



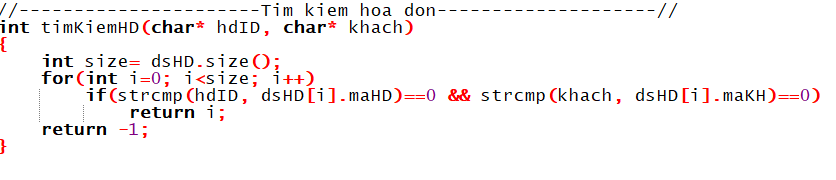
Hình 28. Tìm kiếm khách hàng 001

### 4.3.7. Tìm kiếm hóa đơn (Case 7)

So sánh chuỗi nhập vào, nếu trùng khớp với mã khách hàng và mã hoá đơn thì in ra hóa đơn của khách hàng tương ứng. Nếu không trùng thì thông báo không tìm thấy hóa đơn

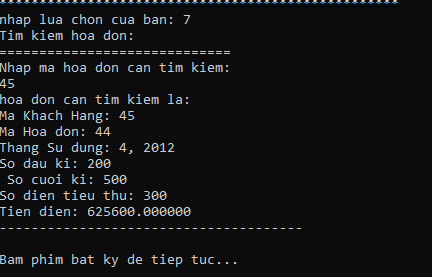


Hình 29.Tìm kiếm hóa đơn



Hình 30. Hàm tìm kiếm hóa đơn

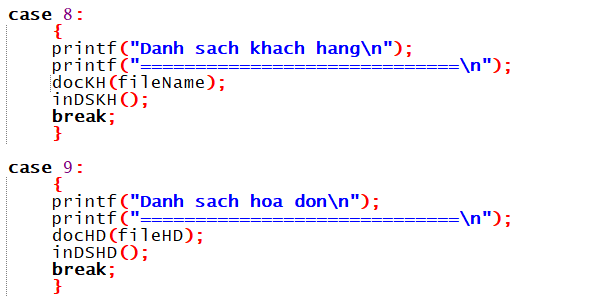
Terminal:



Hình 31. Tìm kiếm hóa đơn khách 45

### 4.3.8. In danh sách khách hàng và danh sách hóa đơn (Case 8 + 9)

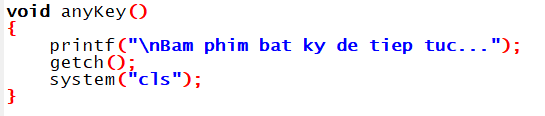
Code:



Hình 32. In danh sách khách hàng và danh sách hóa đơn

### 4.3.9. anyKey()

Dùng hàm này để xóa màn hình, tiếp tục lựa chọn các chức năng khác tùy ý



Hình 33. Hàm anyKey()