EEE101 C Programming and Software Engineering 1

ASSESSMENT 4

­­

**Report­­**

**Authors:**

Yimian Liu: 1717608

Tongxin Niu: 1717286

Xiaotian Li: 1716190

Jingyu Lyu: 1717610

Jiangnan Ye: 1715697

**Content**

[1. Introduction 4](#_Toc533162038)

[2. Specifications 4](#_Toc533162039)

[3. Analysis 6](#_Toc533162043)

[4. Algorithm Design 11](#_Toc533162044)

[5. Implementation 31](#_Toc533162045)

[6. Testing 32](#_Toc533162046)

## Introduction

Nearly all systems with respect to the management are supposed to supported by certain software where human could be able to interact with the systems to take an overall control and master of the status of the specified scenarios, for example the flight management system, library information system, etc., which may be more convenient to operate with the ease of the software. Thus, when it comes to hotel management, the targeted users are divided into various groups and it is vital to utilize the software to take charge of them and view the running state of the hotel, and even further, to help make the hotel director analyze the running situation, producing corresponding financial statement or statistics, assisting manager run the hotel better. The rationale behind performing king of software to assist running the hotel is that it takes full advantage of computer and programming to integrate all the functionality to facilitate the managing of the hotel and improve the efficiency. Moreover, massive and complex projects need teams to complete and negotiate, being assigned to five people. Therefore, this report will contain the part of problem statements(specifications), analysis, algorithm design, implementation (several novel codes) and testing in terms of conference hotel management, respectively.

## Specifications

**2.1 General requirement**

The overall and general requirements for the hotel management is to help monitor the operational condition of the hotel. The function of a hotel, basically, is to provide rooms for customers with different request, deal with the information of customers, arrange different classes of room, make sure the reservation would be accepted, and most paramountly, is in order to ensure the normal running state of the hotel and make money.

* Managing reservations, check in and check out of the hotel which contains 80 rooms, 10 for each floor, and four classes of room (\*\*, \*\*\*, \*\*\*\* and VIP). Each room would be assigned a single price already.
* Manage the account of customer, managers and receptionist.
* Provide the hotel business statistics, such as numbers of VIP customers, average numbers of hotel guests, average and total outcomes, numbers of reservations, room allocation condition, the information of customers, possible future estimate of customer, etc.

**2.2 Functionalities for manager**

The responsibility of a manager, through the system, is to

* Set and ament the price of each class of room
* Set number for each number
* Assign different class to each room
* Determine the number of each classes
* Manage the customer information
* View the running state of the hotel, involving

1. Today’s number of check-in and check-out
2. Reservation condition in the future (days selected)
3. The total and different periods of time’s incomes and its contribution from different classes
4. Number of customers in terms of day, week, month, season and year
5. History information of different classes of room checked in
6. Future assumption of the customers and incomes

**2.3 Functionalities for receptionist**

* Register a booking and check-in customers
* Recording a customer’s name, address, telephone number and hotel member card no.
* Customers without a hotel membership card cannot book a VIP room
* A search facility should be provided for room availability and dates
* The operator can book one or more rooms to a registered customer
* Record the arrival of a customer in the system
* Edit booking details e.g. period of stay or room
* Check-out customers by calculating charges

## Analysis

* 1. **Input**

The analysis of the input will be discussed in two parts: Manager & Receptionist in the order of the functionalities.

* Manager

1. Operating on the interface:

The basic input to operate on each interface

1. Login

Input to choose the identity

Input the password

1. Set class and price

Input floor

Input room number on that floor

Input a choice for class/ input a price

1. View business statistics

Input a choice of which type of statistic the user wants to view

Input a choice of extract the data to Excel

1. Mange a customer database

Input an index of the identity of the customer

Input which operation the user wants to do on the database

Input the change wanted to make on the database

1. Manage the account of the Hotel staff

Input a choice of the operation

Input the password

* Receptionist

1. Operating on the interface:

The basic input to operate on each interface

1. Register & Check in

Input the basic information of the customer （including name, address, phone number, vip number）

1. Search

Input room

Input date

1. Edit booking details

Input the room number

Input the item needing to be changed

Input the new information for that item

1. Check out

Input room

Input date

Input the name of the customer

* 1. **Output**

Several output interfaces would be displayed on the screen to demonstrate the options, corresponding information that is related to the function, and the interaction between the program and the user.

* Menu

The first interface that would be displayed is the menu of the hotel management system that contains selecting different accounts to log in, whether the user is a manager or a receptionist, some notifications, for example the “pressing the Esc” to quit the program, the time, the header of the program, etc. Further, the selection method would be accomplished by pressing the arrow, up, down, left or right to select different options, and the output would be an arrow pointing to the corresponding option.

* Manager

After the user has gained his or her access to the manager account, except the standard output interface, time and header, that has already been discussed previously, the unique selection for the manager would be shown on the screen.

1. Viewing statistics

For example, viewing the income, occupancy rate ,exporting room data to Excel, exporting visitor data to Excel, backing up data to the server online, recovering data from the sever online would be shown on the screen for the manager to choose. Moreover, the special function here is that the Excel would be open for saving and demonstrating the relevant statistics, making it more convenient for manager to perform the data analysis.

1. Account setting

It is assumed that there would only be several receptionists in the hotel and in order to simplify the procedures, all passwords of the account of receptionists are set to the same one. Therefore, the manager could select to change his or her password as a manager, set the password to all the receptionists and clean all passwords of the staff. When the option is chosen, corresponding need of input would be displayed on the screen.

1. Data setting

After the manager has selected the data setting of the hotel, the interface involving several choices would be shown on the screen which are updating room price, changing different room types and updating the guests’ information. When the manager chooses different functions, the corresponding data and requirements would be shown on the screen. For example, manager choosing to update the room price, the next interface would display the room price current and require for manager to input different classes’ prices.

* Receptionist

There have four main functions arranged for receptionists. When user logs in as a receptionist, the interface would demonstrate four options for receptionist to choose. One it to book room for the customers, after the receptionist choosing this option, system requiring user to input several information to finish the procedures of booking one room. The second one it to update users’ information, updating users’ information with several information. The last two functions are check-in and check-out in which the reception would input the customers’ information and sum up all the fees that customers may consume.

## Algorithm Design

* 1. **Hiding Password Display**

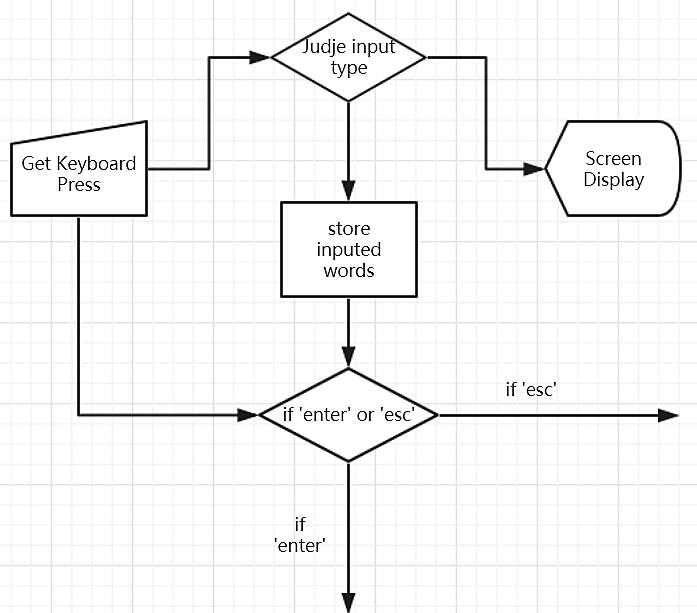


Figure 1

* **Objective:**

This algorithm is designed to hide the password that would have been displayed on the screen and replace them with signs (such as “\*”) displayed. As a result, others could not see the user’s password when inputting, and the security of the program will be greatly enhanced.

* **Principle:**

Each time, the program will first examine and get one keyboard press from the user. Then, because this is designed for password inputting process, the program should examine the type of the input to make sure each input is only the part of number or letter or signs on the keyboard. If the item that user inputs is valid, the program will display one sign (“\*”) for each item and store it in an array.

Simultaneously, when conducting the steps above, the program will detecting the press of “Enter”and “Esc”on the keyboard.If “Enter”is pressed at any time, the program will think of it as the indication of complete input process for the user and run into the next interface of procedure. If “Esc” is pressed at any time, the program will return to the last procedure or interface and the password input just now will not be stored to next time.

* 1. **Data deletion**

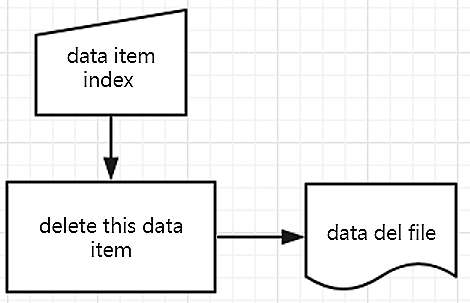


Figure 2

* **Objective:**

This algorithm is designed to delete data that the program does not need. For this program, it must be available to store a significant amount of data of visitors, for example, their ID numbers, nationality, check-in history. Once data needs to be deleted, the program will do the deletion execution. However, for C language, it is confirmed that data is easy to be stored and added, but it is difficult to delete, because the data cannot be deleted actually. What is be done is just hiding the data. What should be done is let compiler know what should be hidden and do not print it out.

* **Principle:**

Once the program runs the deletion execution, users are deciding what should be deleted. Then users can use data item index to choose the data they want to delete. Once the data item is locked, the item is moved to data del file as Figure 1. shows. Once the data needs to be printed, the program will scan the data del file first to check if the data is matched in data del file. If it is matched, then the program does not print out the data that is matched in data del file. In fact, the data are not deleted, they are just hidden.

* 1. **Data Fetching**

This algorithm is combined with two partial algorithms: Examine the Harsh Table, Locate and Fetch

* + 1. **Examine the Harsh Table**

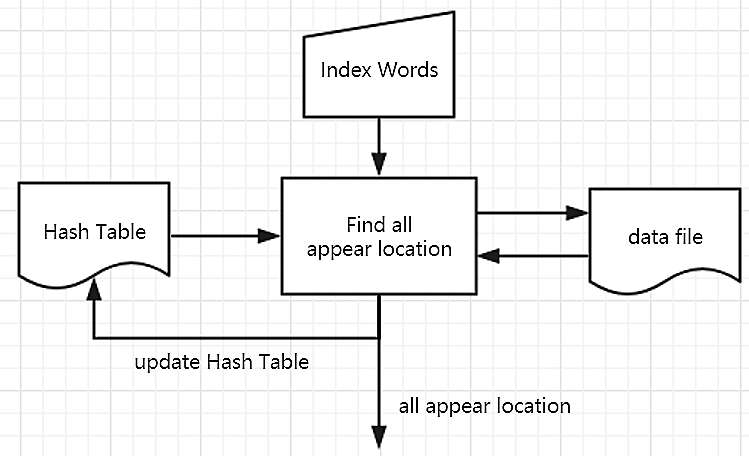
****

Figure 3

* **Objective:**

By setting the Harsh Table which store the **address** of the data of the same type, the searching time of the program will be greatly reduced after the first searching process.

* **Principle:**

Before fetching the data, what type of data needed to find should certainly be determined, so we can hold what we want to search it in the file. That is called the ‘index’—— the type of item we want. When first searching the file, the program will first find all the **addresses** of all the items with the same type of the index. Then, the program will read the data from these addressed and compare them with the index we want. Simultaneously, the **appear** **address** of all the data with the type of the index will be stored in the Harsh Table. Next time the program conducts the searching process, it will first **examine the Harsh Table** to see if this type of index has been searched before. If not, the program will do the search as the first time. If yes, the program won’t search the whole file. First it will get these from the Harsh Table for search. Next, it will continue searching but not start it over from the beginning of the file. It will search from the **last address of the same type of the data recorded in the Harsh Table.** If it finds new address for this index, it will add them to the Harsh Table to refresh it and get these for search.

* + 1. **Locate and Fetch**

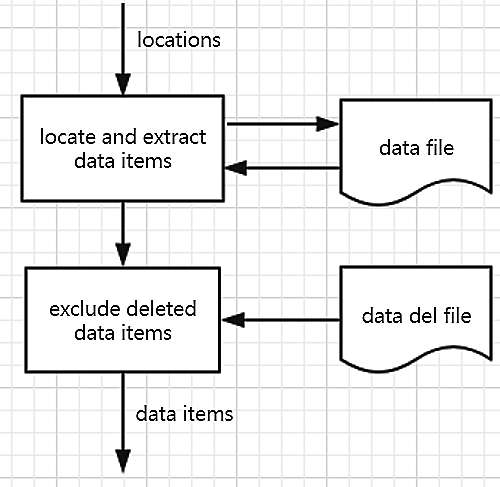


Figure 4

* **Objective:**

This algorithm is designed to cooperate with the Harsh Table to fetch the data wanted with the shortest time.The data deletion file has been introduced in the Data Deletion Algorithm, so the fetching process will avoid getting these data in that file to reach the aim of deleting the data.

* **Principle:**

The program will receive two sets of address from the Examine the Harsh Table Algorithm. Next, the program will fetch the data from these addresses. By comparing with and excluding the data in the data deletion file, the program will finally get one set of data items of the index and return that final result.

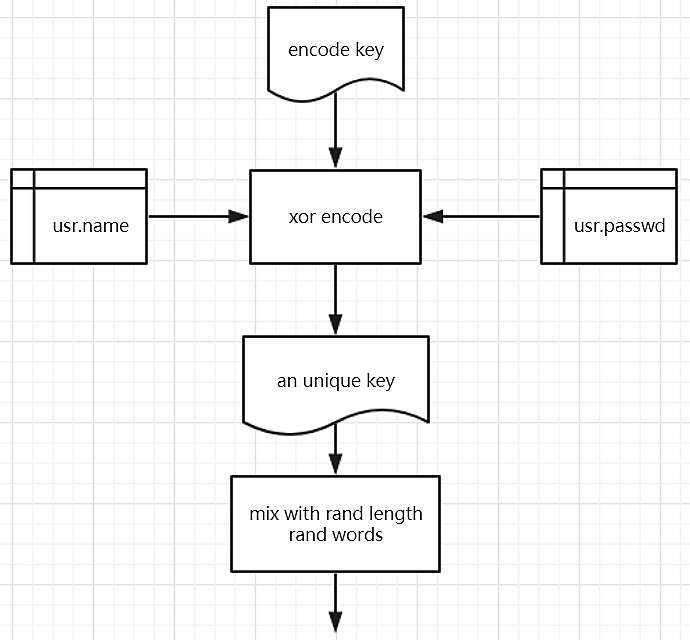
****

Figure 5

* 1. **Encryption of Password**
* **Objective**

The input password of all users would be encrypted to secure the information of customer in order to mimic the real world application of hotel management that each customer or inner staffs’ account information are extremely vital. The algorithm design of the encryption off the password is displayed in figure 1.

* **Principal**

The method of xor encode is one of the innovative approaches utilized to encrypt users’ information. When the user has input his or her password and account information for the system to save them, the encode key that is designed and known by the programmer acts on the input information to transform them into the strings that is total different from the former one. The basic working principal is to change some of the binary number, 1 or 0, in the saved information with a standard rule. For example, the original binary number of an integer is 1010, the specified encode key transformed or 1 into 0 and change their sequence, in consequence of 0110. Thus, a new integer has been created by the transformation of the this encode key which is a unique key of strings that only the accessor and programmer could know the original strings that are input. At the last stage of the encryption of password which could been seen as the portion of users’ input, the new strings that has already been transformed with the encode key would be mixed with other characters and words with different length and then been put into different position in one particular string. For example, the unique string that has been created after the effect of the encode key is a8k7, more complex in the real application, there only a simple example. After the mix with random length of characters and words and changing their sequence, the final password been saved in the file turned into 6kt8a1m7 in which first changing the original sequence of the password a8k7, into k8a7, then adding the new characters and words, 6t1m, between them, finally becoming 6kta1m7 which is saved from being stolen and data theft, making the all accounts secured to some extent. The encode key and the mixture with new words and characters would not been the same as the example discussed before that is only made it simple to convey how this principal of encryption of information works.

When the programmer and the designer of the system want to access the data information and the particular information of the user, the methodology used to decode the unique key string is the same as encrypt it. The unique encode key is known and charged by the system manager and when they superimpose the encode key to the string that is saved in the file, that could be able to get the authentic original input password that user has input before, securing the accounts’ information.

* 1. **Digital Signature**

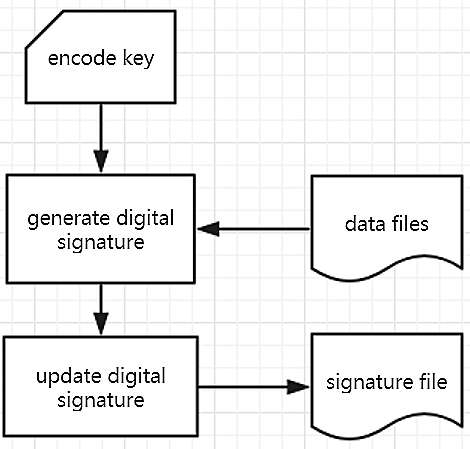
**

Figure 6

* **Objective**

Given that the security of users’ information, from other aspect, after finishing the input of users’ information, the malicious tampering with the file is supposed to be avoided when designing the program, utilizing the digital signature marked with one particular string for each file.

* **Principal**

The digital signature of the specified saved file would be generated to mark the data files in which contains a large amount of information generated when dealing with the input and output of the information. Further, one particular file that saves the information would generate one single digital signature that distinguish it with each other file that saves the information. After the information of the original information in the file is changed due to the input and modification of user, the digital signature would be updated automatically and generate a new signature file that only saves the string of the digital signature file. For instance, one user’s information is saved as a structure in a particular file, and the corresponding file marked the information that is saved as the binary numbers of the file generated is called the digital signature, with only one string. The advantage of using this method is to mark one file containing amount of information and then it could be checked whether or not the original information in that file is modified by someone else because once the information in the file changed, the digital signature would be altered, correspondingly.

* 1. **Judging the authenticity of the original information**

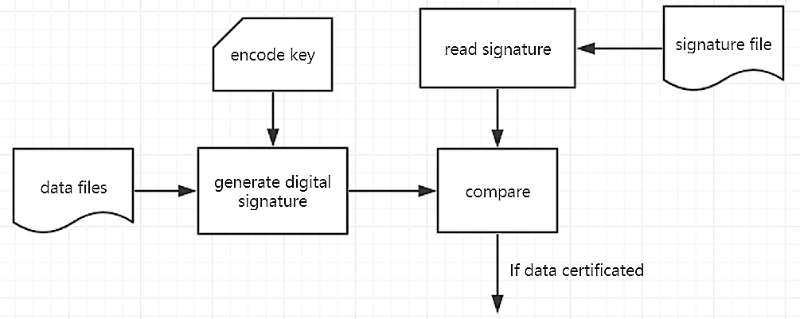
**

Figure 7

* **Objective**

It’s Similar to the algorithm before that has already created a new signature file for each original file that contains the file, the method of checking whether the information in the file is changed or not is used to secure accounts’ information in the whole system.

* **Principal**

The first step, in order to compare the two signature file names, is, still, use the encode key to operate on the data files which is new to the system and then a new digital signature of that file would be generated accordingly. After that, the system would read the signature in the signature file that has been generated before and compare these two digital signatures. If both two of the digital signatures are same, it is implied that the information opened and viewed by the manager of the system is the same as the last time he or she did it. However, if two digital signatures are not the same, it could be deducted that the information in that particular file is changed by other hackers, reminding the system manager that that information in that file is not dependable anymore and the encryption of the information should be enhanced.

* 1. **Keyboard monitoring**

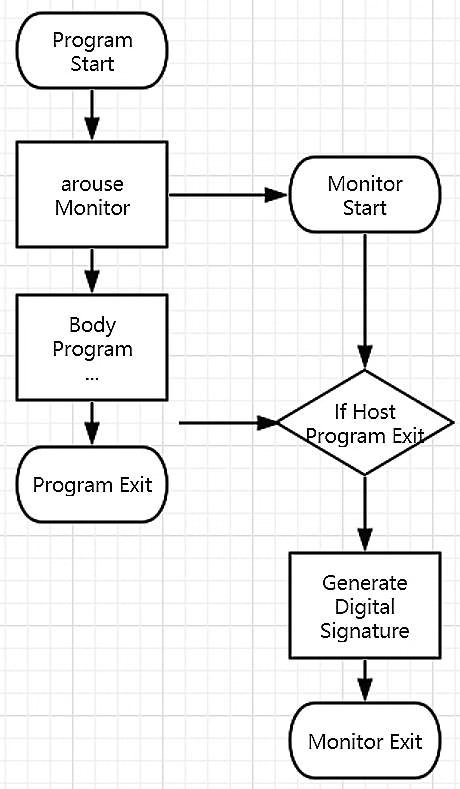
**

Figure 8

1. Objective

In order to generated the file that is particular referring and marked with one special and unique string, the keyboard monitoring mechanism is used to cooperate with the ease of digital signature when saving the file.

1. Principal

In parallel with the main program running, a sub-program is set to perform synchronously. For example, when the exe document is open to run and the hotel manage system is activated, one another program is activated simultaneously. However, this parallel program would not be demonstrated in the user’s interface and is run background. The only demonstrating interface shown to the user is still the main function of the program. As long as the user of the system has entered the “esc” button to quit the corresponding function that is related to the information saving and file, the keyboard monitoring sub-program could detect this enter on the keyboard and is able to generate the digital signature file at the same time. Quite similar to click the quit button of any one executable program that the program would be terminated after the clicking, the keyboard monitor takes the similar mechanism that once it detects the “quit” button on the keyboard, the program would perform several actions, particularly in here, referring to generate the digital signature. The utilization of the keyboard monitoring methodology is to coordinate with the use of digital signature. The reason why using this method is that the program would not know when to generate the digital signature of the specified file and in order to achieve that function, the keyboard monitoring method is introduced to handle this problem. Further, due to the parallel running of this sub-program with the main function, it would not affect the running speed of the original function is main(), speeding up the running space of the program.

* 1. **Arrow Interface interaction**

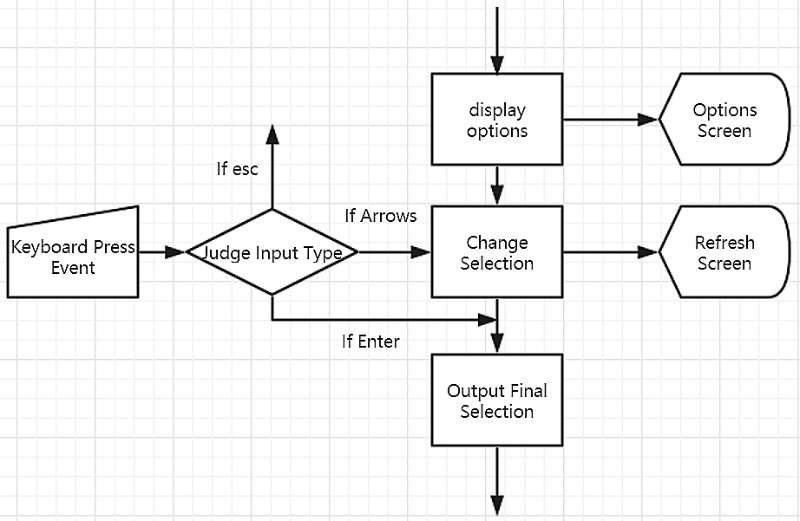


Figure 9

1. **Objective:**

The program deserts the complex way of operating on the interface by inputting the number in front of the choice to get in the functionality we want. Instead, the program uses the direction keys on the keyboard to control an arrow on the interface to interact with the interface which not only makes it more convenient but also adds entertainment for the user.

1. **Principle:**

First and each time, the program will get the input from the keyboard and examine the input type. If the input type is ‘Enter’, the program will run to the function that the arrow is pointing at. If the input is the direction keys, the program will change with the selection by cleaning the interface to print the arrow sign on the new selection, also other options. The key point on this is the usage of remainder. If there are 3 options, then we can use a relatively large number ‘N’ to divide 3. As a result, there will be 3 possible remainders (0,1,2). By adding 1 to each remainder, we can get 3 possible number (1,2,3) that is different by 1. Every time the ↑ or → button on the keyboard is pressed, the program will return 1. Every time the ↓ or ← button is pressed, the program will return by -1. By add each press value that returned to the large number N, the remainder will also be changed with +1 or -1 which forms a loop of (1,2,3) of the remainder. By this method, the arrow will be continuously printed according to the value of remainder as if the user is moving the arrow. Additionally, it is worth mentioning that the initial value of the large number should be set to what can exactly divided by the number of options.

* 1. **Excel and file**

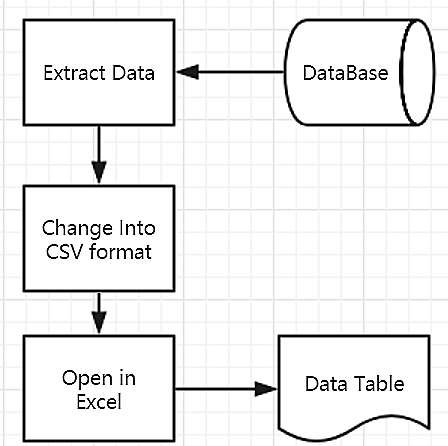


Figure 10

* **Objective**

This algorithm is to print the data out and export data to excel. This allows mangers to check operation of the hotel. The program is required to print data about room IDs, types, prices, costumers, and check-in or check-out in 7 days, it can be imagined that it is difficult to read data. Here is a picture to show some pieces of the whole data in **Figure 10**.

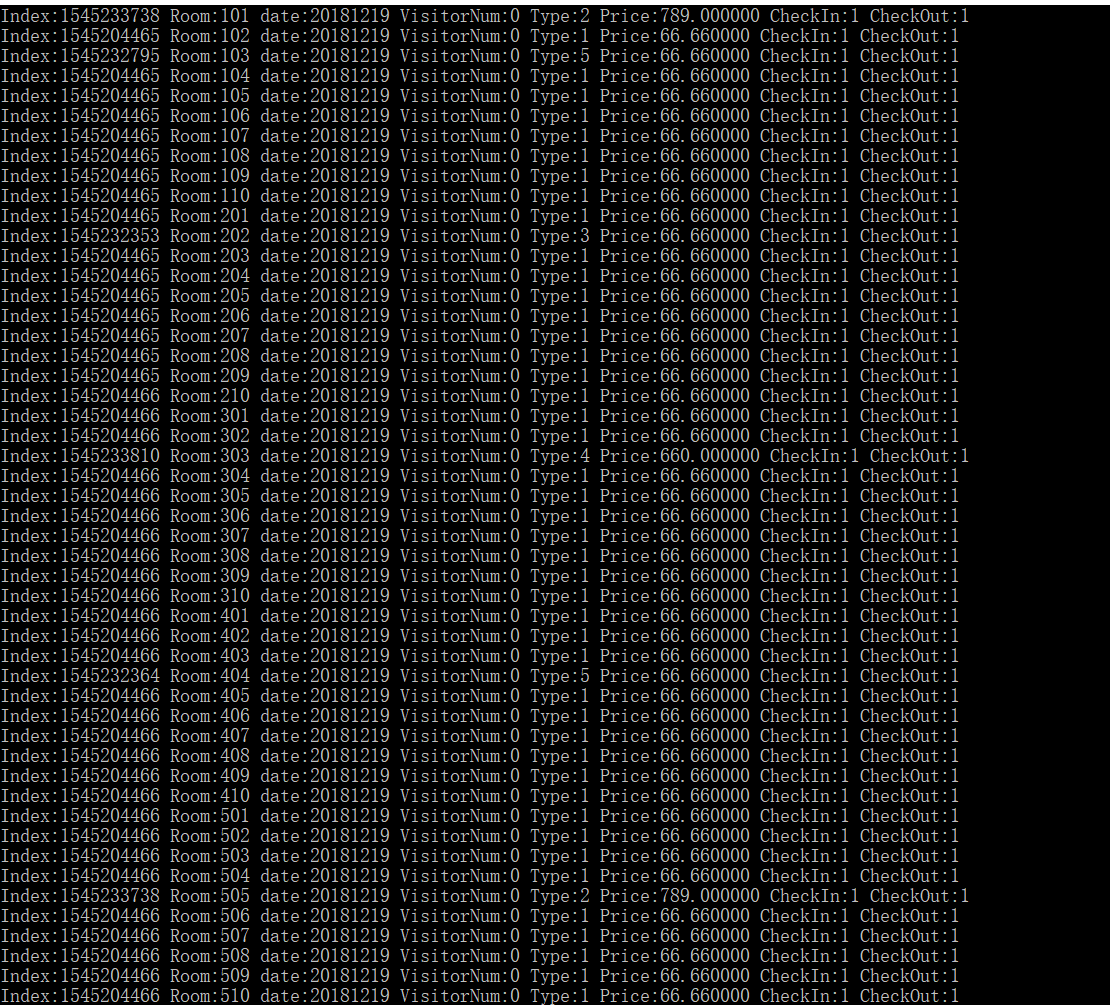


Figure 11

However, this algorithm can export data into excel, and data in excel is shown like this **Figure 11.**

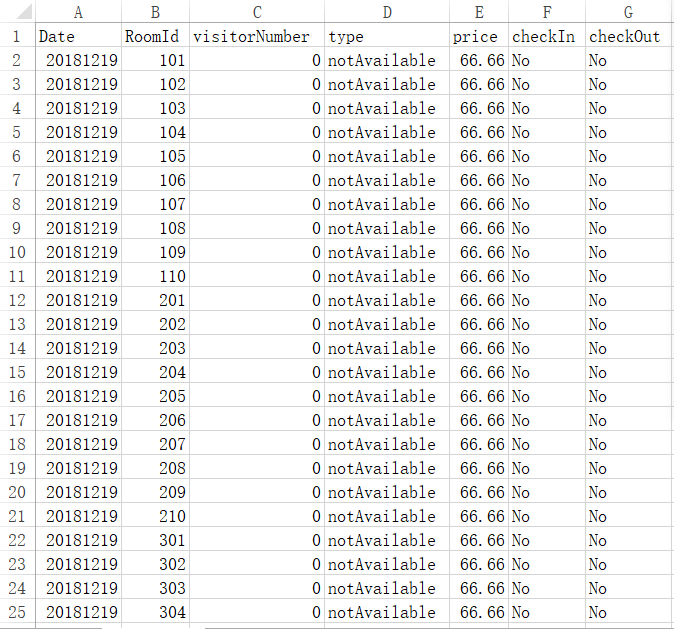
****

Figure 12

Data printed in two figures is same, mangers can check data in **Figure 11**. It is much easier to read than it is in **Figure 10**.

* **Principle:**

When the program runs, all data is stored in the strings in hash table as mentioned before. When the program does this execution, it checks the special symbol for each particular data item to match the data wanted. Once data is matched, then it will print data out and build in “.csv” format. Then system open the Excel, data would be transported into Excel.

* 1. **Multithreading**

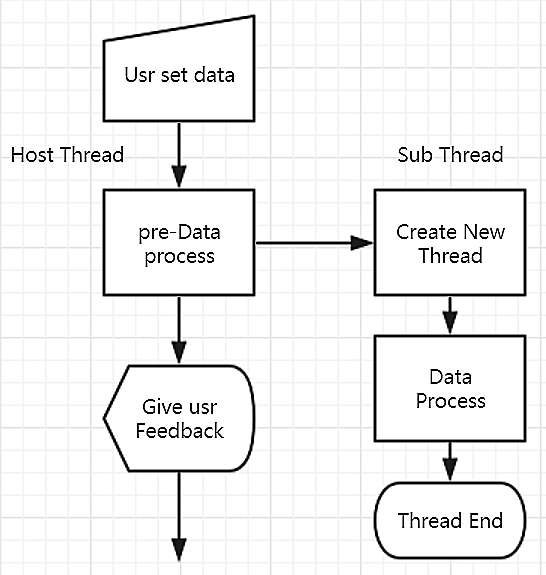


Figure 13

* **Objective**

This algorithm is designed to increase the program speed of handling data. As mentioned before, there are 80 rooms for ten floors. A large amount of data needs to be stored such as IDs, types, and price. Besides, visitors‘ data such as booking information check-in or check-out and ID numbers needs to be stored, too. Furthermore, data in 7 days also needs to be stored. If all this data is stored by using one dominant threading, it must be very slow to handle all data. Visitors, mangers and receptionist could be annoyed if computers work so slowly. So secondary threading is designed to store data and future information. For dominate threading, it has other work to do, which is showing interfaces to mangers and receptionist. This means the whole task is divided into several small tasks and distribute them to the programs. It increases handling data efficiency.

* **Principle**

When the data needs to be set, the program is running according to Figure 5. For the host threading, it handles pre-process data and does preparation execution such as check-in or check-out and room booking for 1 day. Then it gives mangers or receptionists feedback immediately. For the secondary threading, when mangers or receptionists input data, it is running to store much more complicated data, the work is done. These two pieces of threading can run together. Because storing data would take some time, and users need to be replied presently, this algorithm can solve this issue perfectly.

## Implementation

**Main.c**

Include the program body.

**Data.h**

Include functions dealing with data.

**Input.h**

Include functions taking charge of Input Action.

**Print.h**

Include functions displaying the screens.

**Login.h**

Include functions controlling login.

**Download.h**

Include functions to download a file.

**Menu.h**

Include functions concerning menu logic.

**Maincode.h**

Include functions of main.c logic.

**Demo.h**

Include demo function for teaching and debug purpose.

**Github**

<https://github.com/string1995/ee101-as4-hotelsystem/>

## Testing

The testing process will be conducted in 4 large parts: manager, customer, data processing part and Home page part. Each functionality will be carefully tested in this section.

* 1. **Data processing**

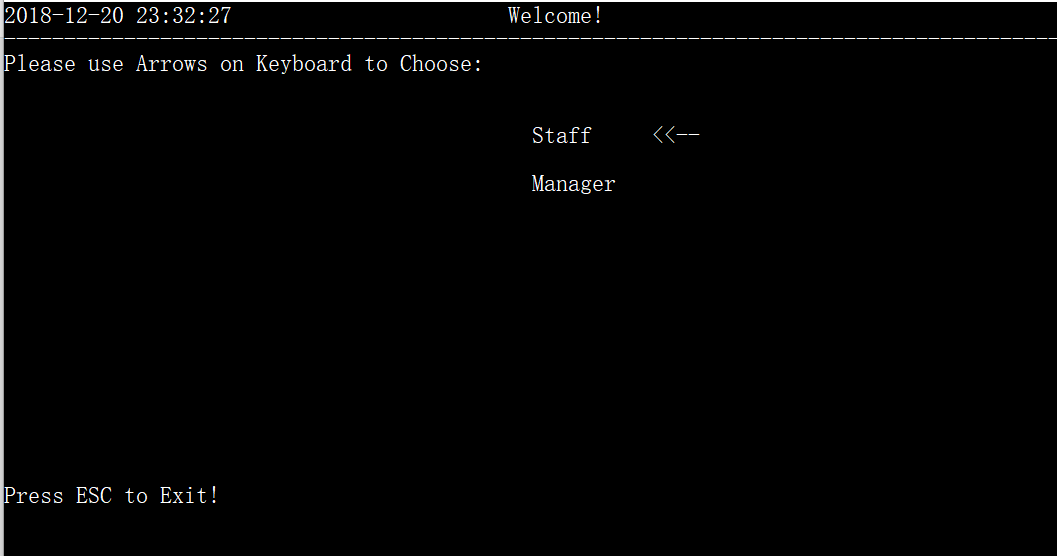
1. **Running speed**

For running speed, it is mainly considered data storing speed because other functions like printing data out run very fast. Another reason why data storing speed is considered is that a large mount of data needs to be printed out or be operated like storing, changing and deleting. The point is that this program allows users to change batch data. It is easily thought once a data item was changed, all data items which relevant it would be changed. For creating data, it was finished in 1 minute if it needs to be created 8 days data 640 items into a file. But if it needs to be created few items, it was finished almost instantly. On the country, speed of changing data is high, because extracting data is fast. It was finished instantly by testing.

1. **Stability**

For stability, it was tested for 50 times, it never crashed. Program was executed from beginning to the end in 20 times. For the rest 30 times, it was stopped in different parts. Whether it was stopped by executing quit set in the program or stopped by closing the compiler, it always kept working after opening again. Besides, although data was stored before program closed, data stored before could be printed out. This program works perfectly, and it is almost impossible to crash.

1. **Multi-thread Data processing**

****

The initialized interface has been displayed as seen in the picture above

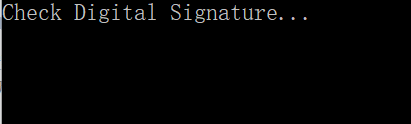
**C:\Users\叶江南\AppData\Local\Temp\WeChat Files\72267aae4d29a9690c55af647f24841.png**

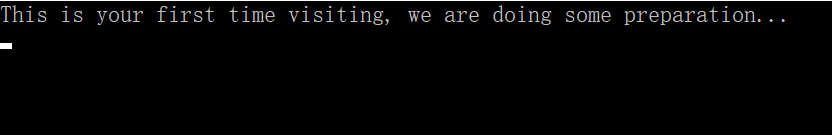
**C:\Users\叶江南\AppData\Local\Temp\1545348775(1).png**

However, the data is still initializing because that the size of the file is still growing as seen in the picture.

* 1. **Home Page**

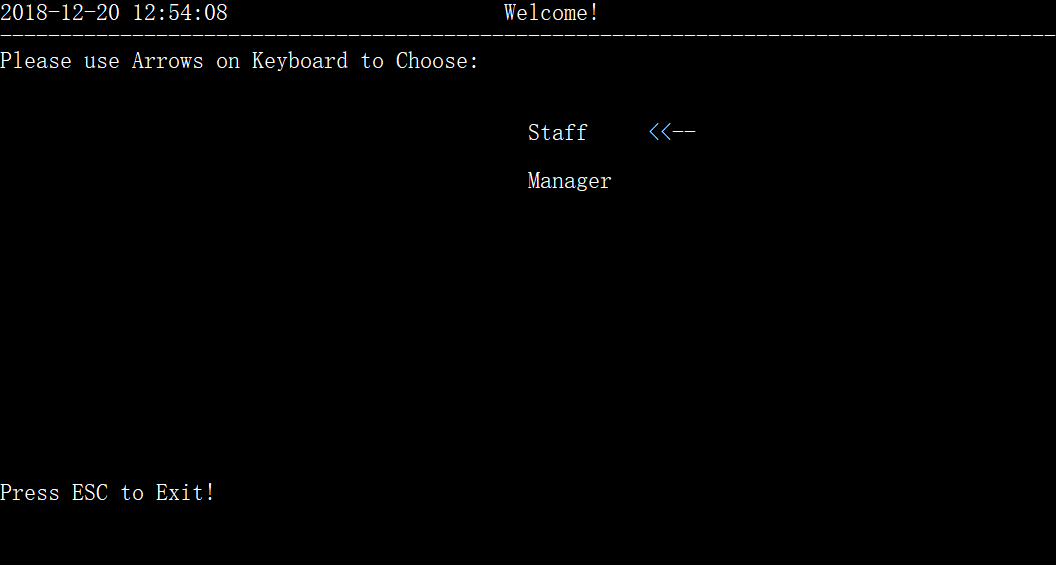
1. **When first opening the function, it needs some time to preprocess.**





The program expectedly provides friendly reminding message to informing the user to wait with patience.

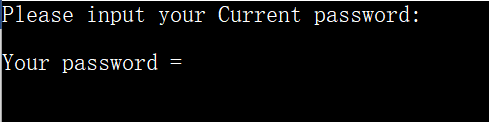
1. **Test the home page**



The home page is displayed with the operable sign of “arrow” which is as expected.

* 1. **Manager**

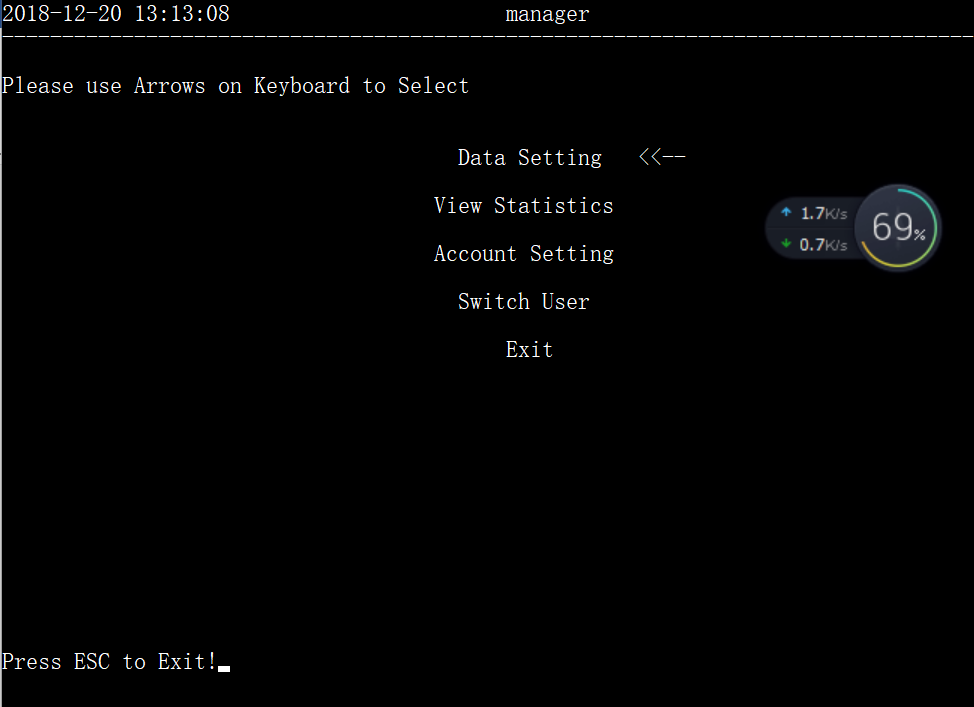
1. **Enter the “Manager” option**

****

the system requires password. The default password is set “123”.

**C:\Users\叶江南\AppData\Local\Temp\1545311239(1).png**

When entering the password, the password item is hidden with “\*”as expected.



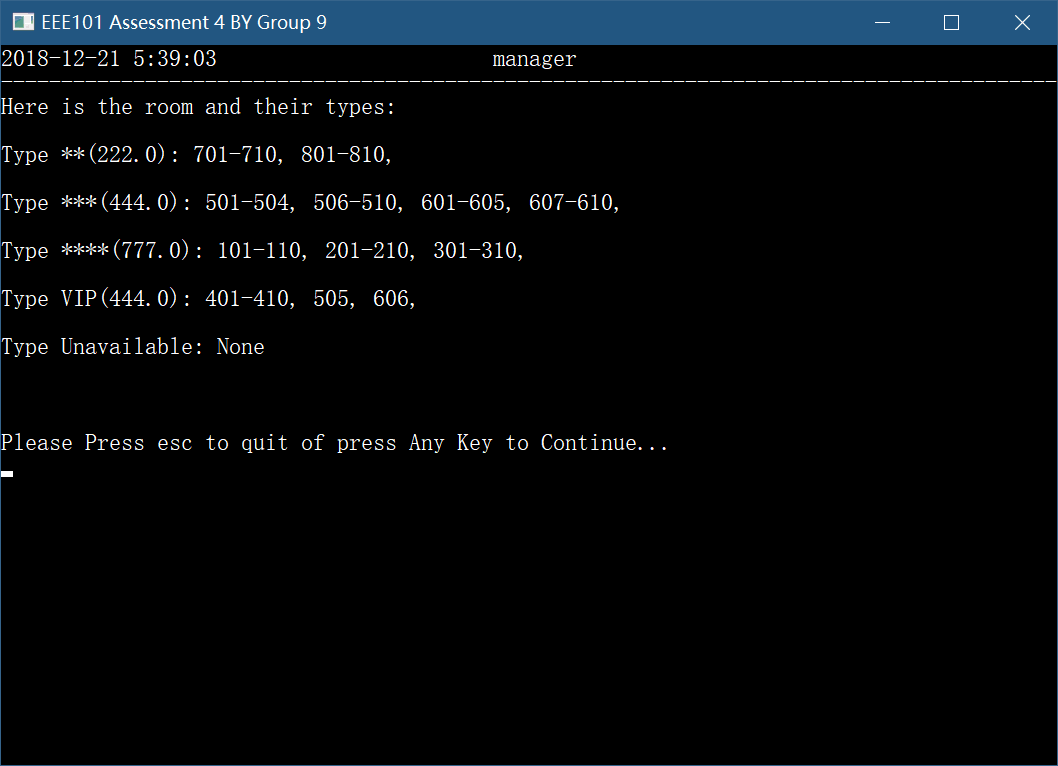
When the password is examined correct, the next interface is printed as expect with the title changed to “Manger”.

1. **Enter “Data Setting” option**

****

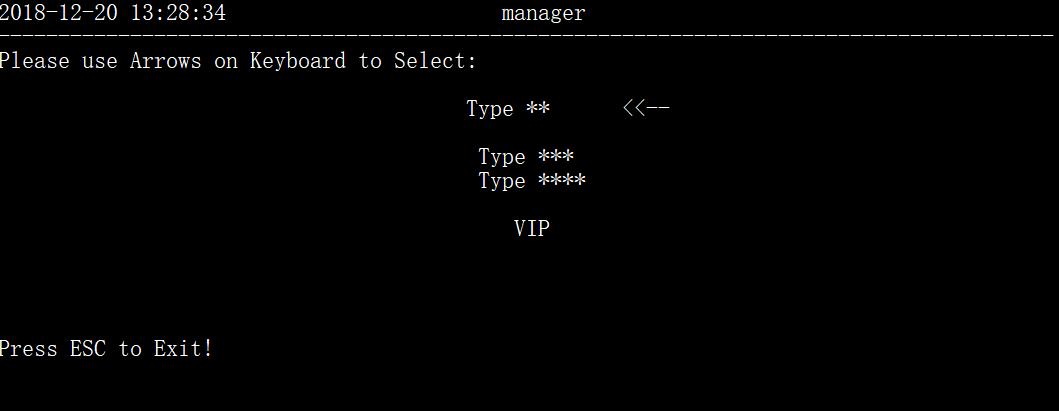
The next interface is printed as expected.

* 1. **Enter “Update Room Price” option**



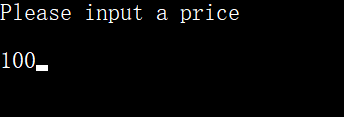
All the rooms distinguished by type marked with price in the bracket are listed as expected. It’s worth mentioning that all the data are sorted in order. And the program waits until the user presses a key to continue.

* + 1. **Select a Type**

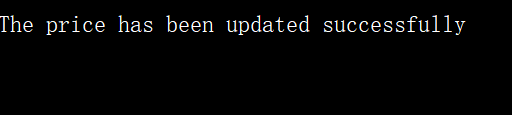


After the user press a button, the next interface is displayed.

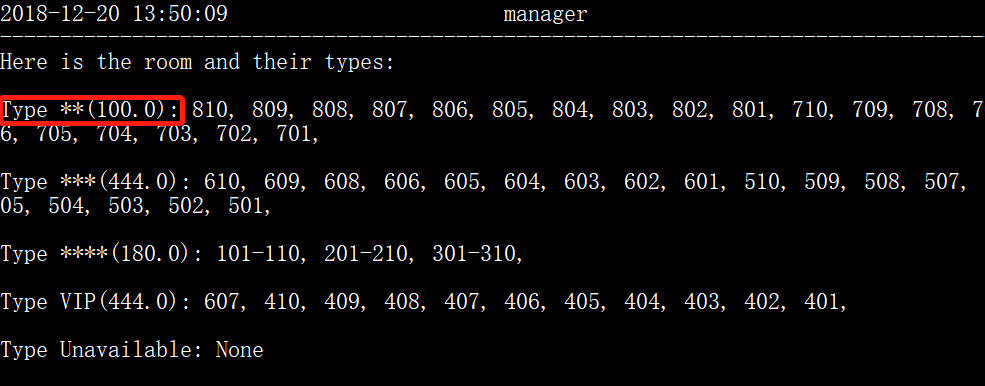
Enter the first type:



The program expected asks the user to input a price and 100 is input for testing.

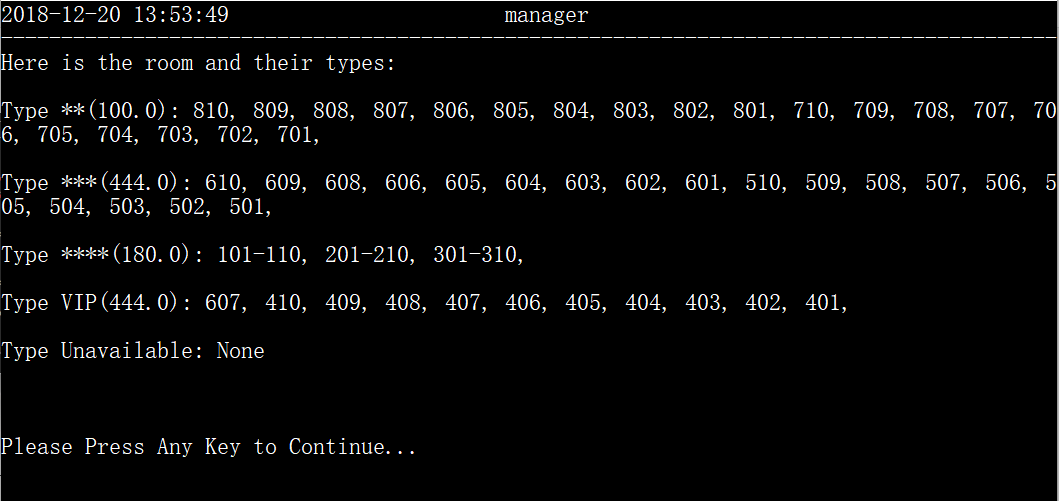


Then the reminding message is printed to tell the user successful price updating.



When checking the price again, the relative item has been changed in the table.

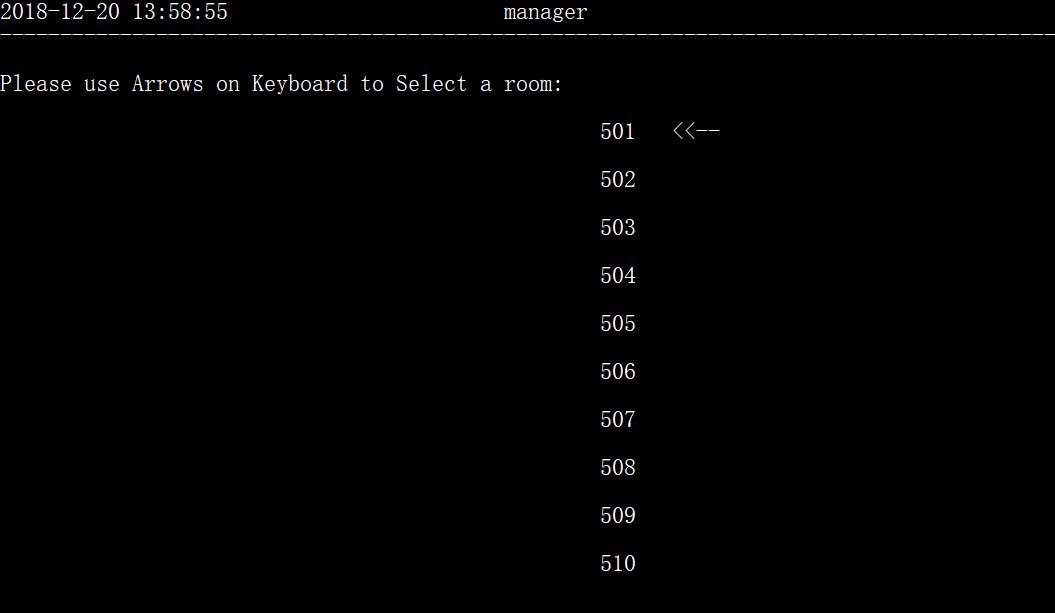
* 1. **Enter “Changing Room Type”**



All the rooms distinguished by type marked with price in the bracket are listed as expected. And the program waits until the user presses a key to continue.



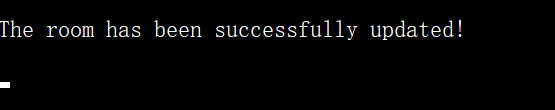
After pressing a key, the next interface is displayed and the user is told to select a floor as expected.



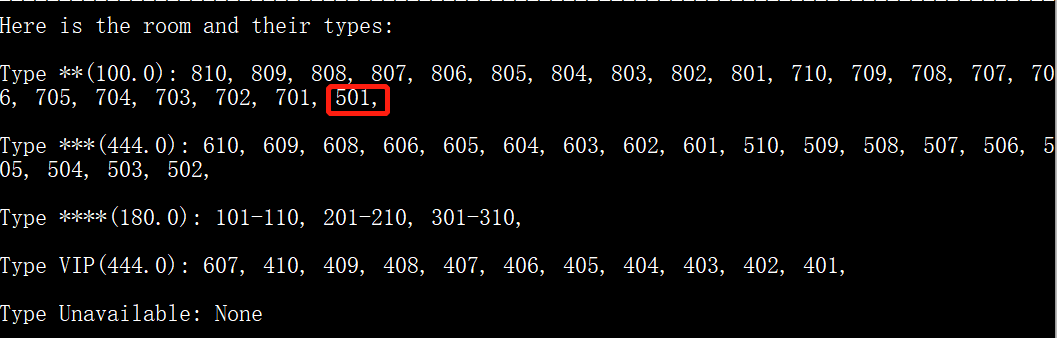
After selecting a floor, the room number on that floor is displayed as expected.



Then entering a ‘room’, type is the next to be selected as expected.

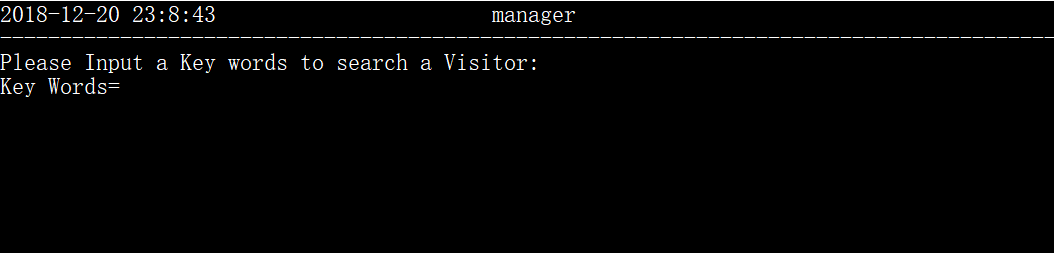


Selecting ‘501’ for testing, then the program prints reminding message indicating successful updating.

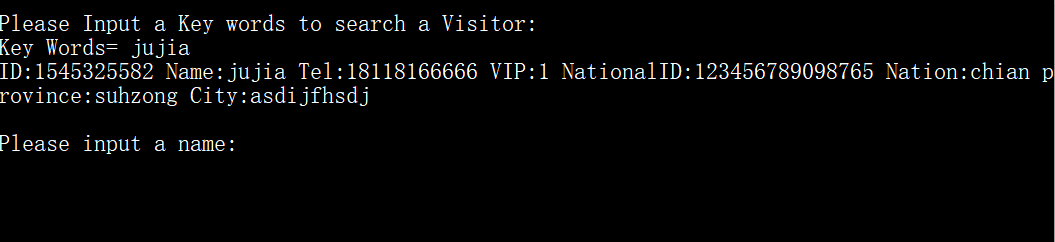


Next time viewing the type table, ‘Room 501’ has expectedly been added to the revised type.

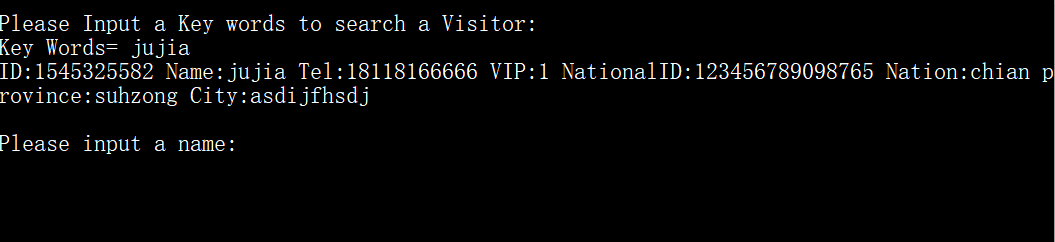
* 1. **Enter “Updating guests’ info”**

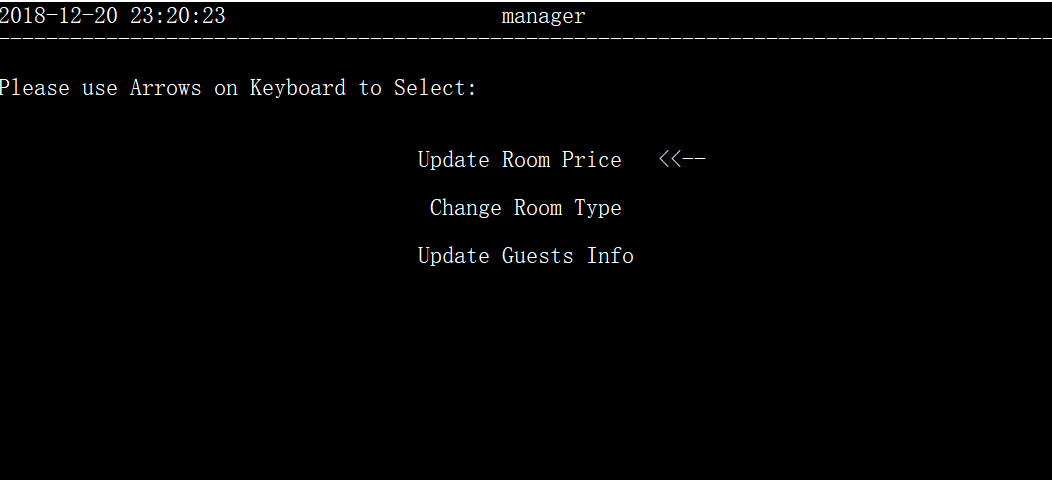


The program asks the user to input an index for searching at first, which is as expected.



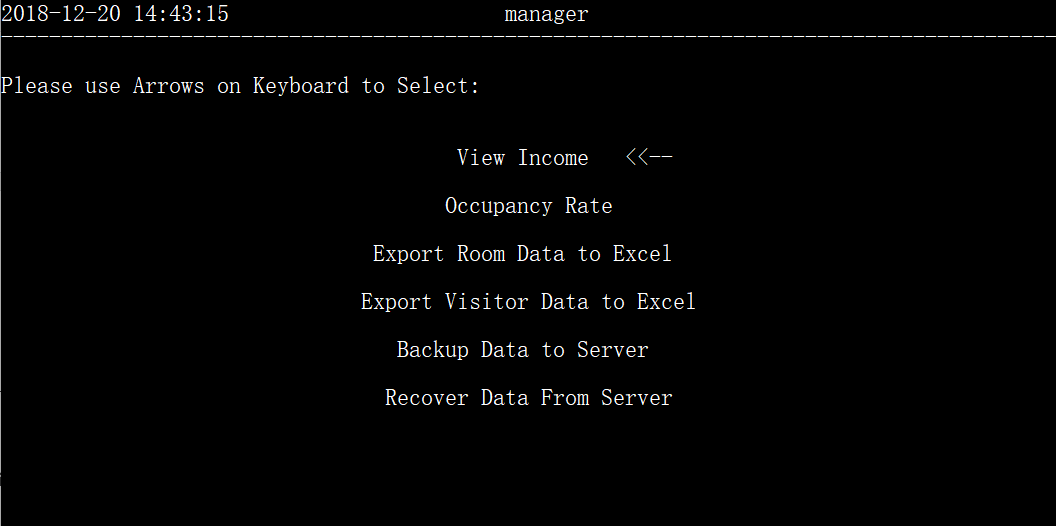
When a keyword (index) is input, the program immediately finds the item containing the index and next asks the user to input the new information of the guest (including name, phone number, national Id, and so on).





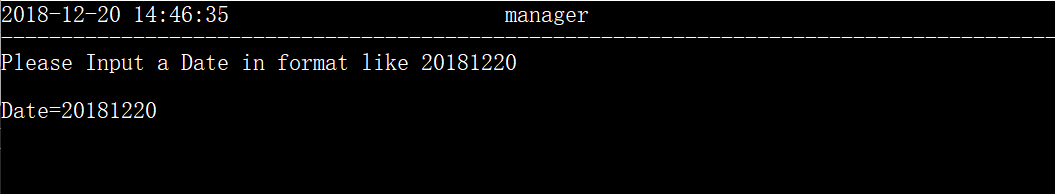
Then, the program returns to the last interface indicating successful updating.

1. **View Statistics**

****

The next interface is printed as expected.

**3.1) Enter “View Income” option**

****

The user is required to input date as expected because the income condition can be viewed correct to date, and 20181220 is used for testing



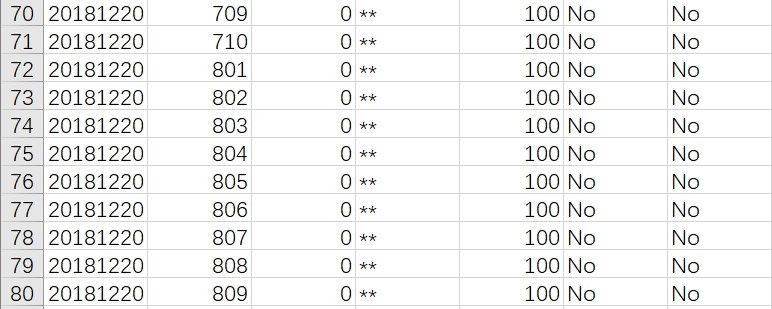
The income is listed as expected (all the result is 0 because there is no register today).

**3.2) Enter “Occupancy Rate” Option**

****

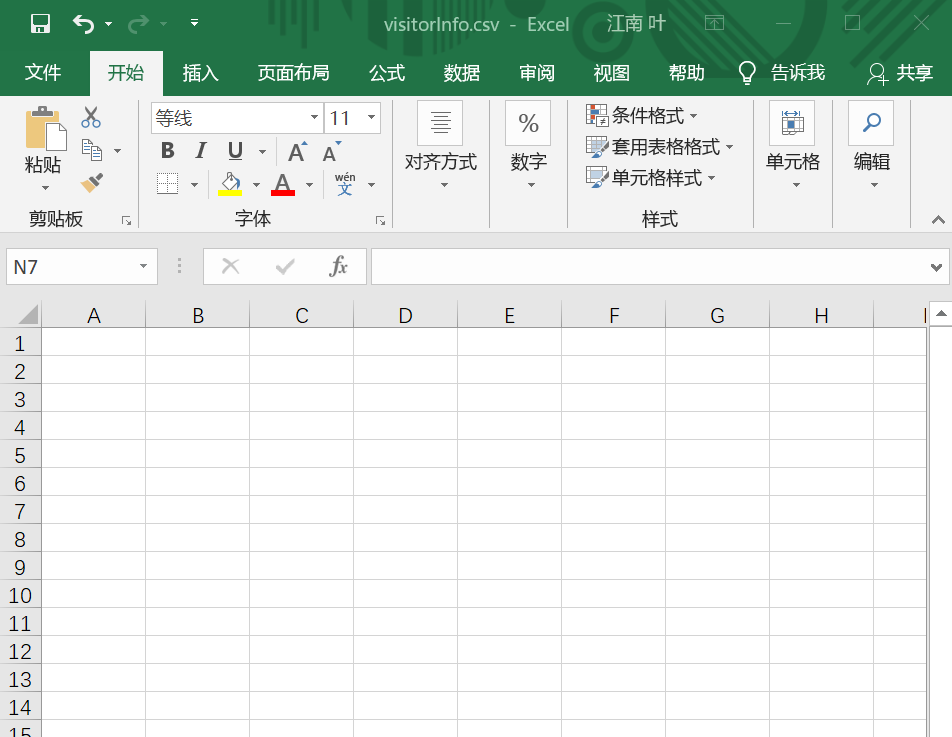
The data are displayed as expected ( because there is no one check in today, the rate all zero percent.

**3.3）Enter “Export Room Data to Excel”**

****

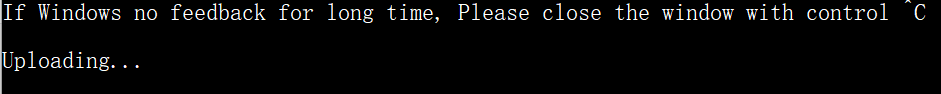
An Excel file is automatically opened. Inside this file are the room information including the rooms whose prices are set to 100 as shown above.

**3.4) Enter Export Visitor Data to Excel**



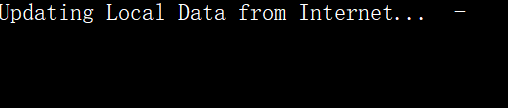
An Excel named “visitorinfo” is created but empty because there is no one check in yet.

**3.4) Enter “Backup Data to Server”**

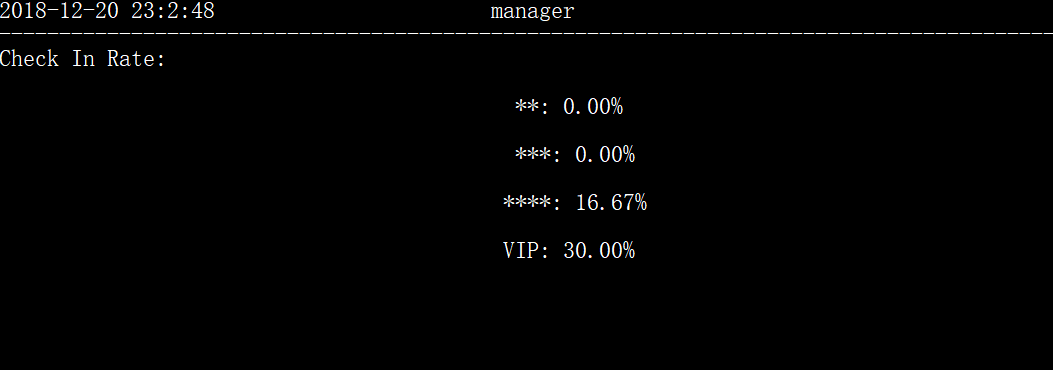


The program displays loading message for the user ,and when it returns to the last interface, the data are backed up.

**3.5) Enter “Recover Data from Server” option**



The reminding message of updating is printed as expected.

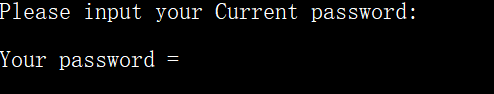


Then when viewing the check in rate, the foregone data are again displayed which proves that the data recovering process is valid.

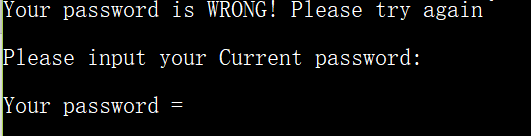
1. **Account setting:**

* Test 1: (change my password)

When the first function of the manager, change my password, is chosen by clicking the enter button. It is demonstrated as the figure below.

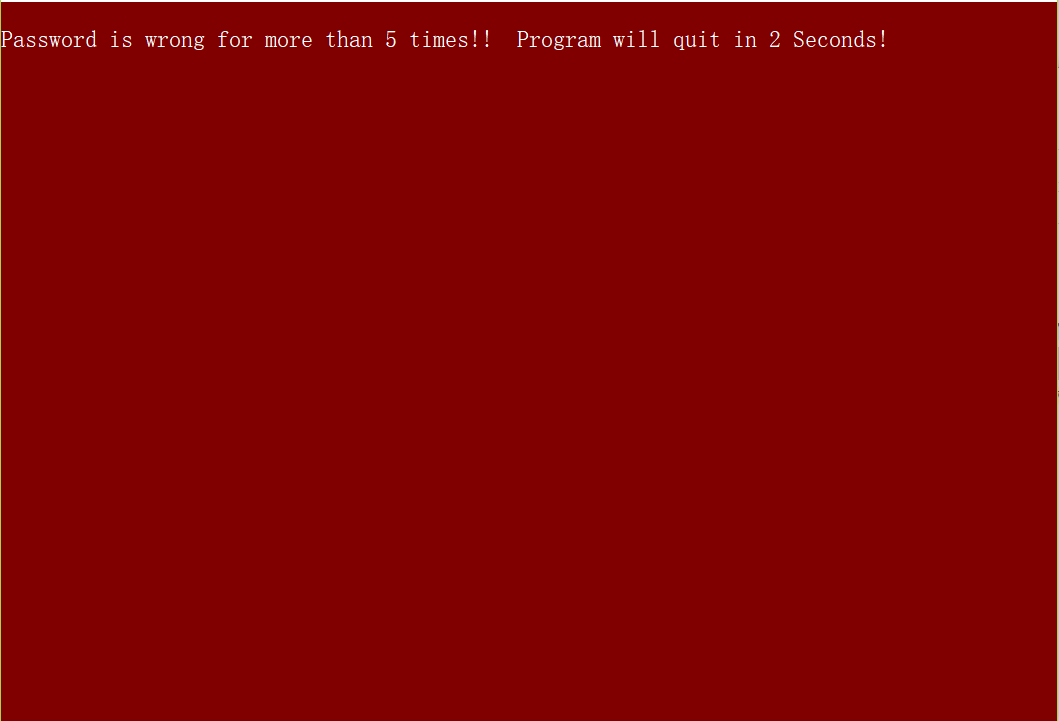


* Test 2: (input the wrong current password)



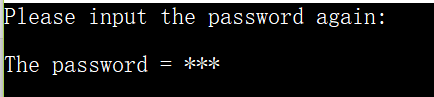
It is reminded that the input password is wrong and program require the manager to input the password again.

* Test 3: (wrong input of password more than 5 times)



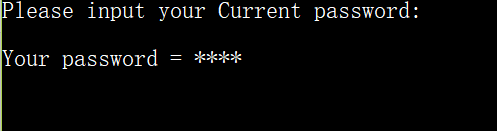
The program would detect the wrong times of the input of password. If is overpasses the 5 times, the program would terminate in 2 seconds.

* Test 4: (doubling the input of new password)



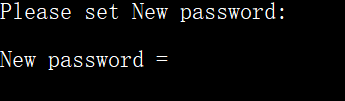
It turns out to work well.

* Test 5: (create staff password)



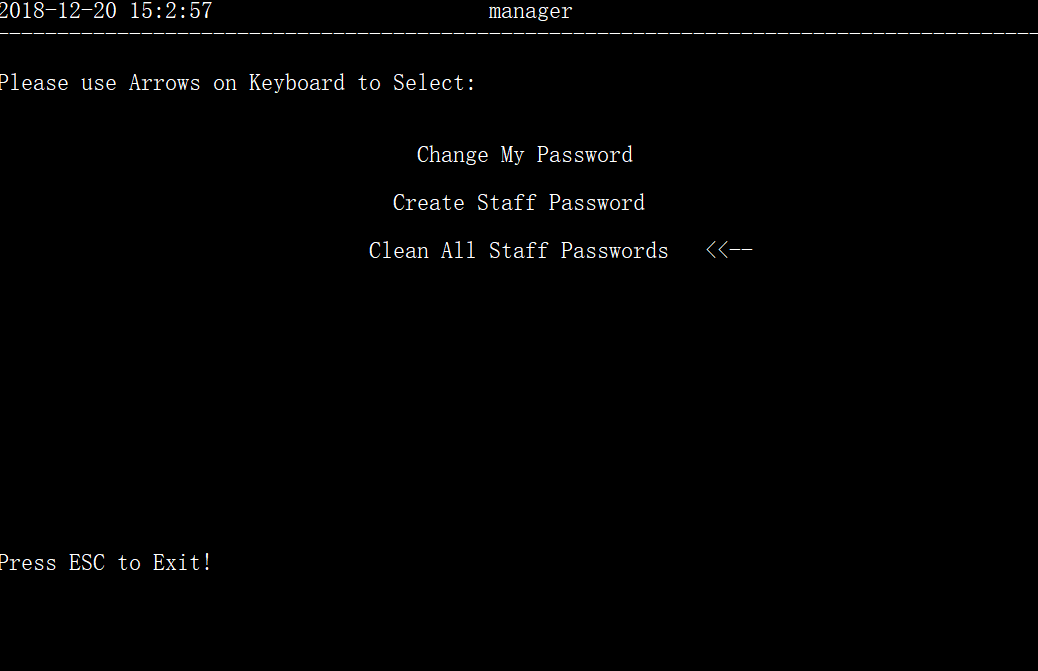
Similar as before, the program requires the manager to input his or her current password to double ensure the identity to secure the system.

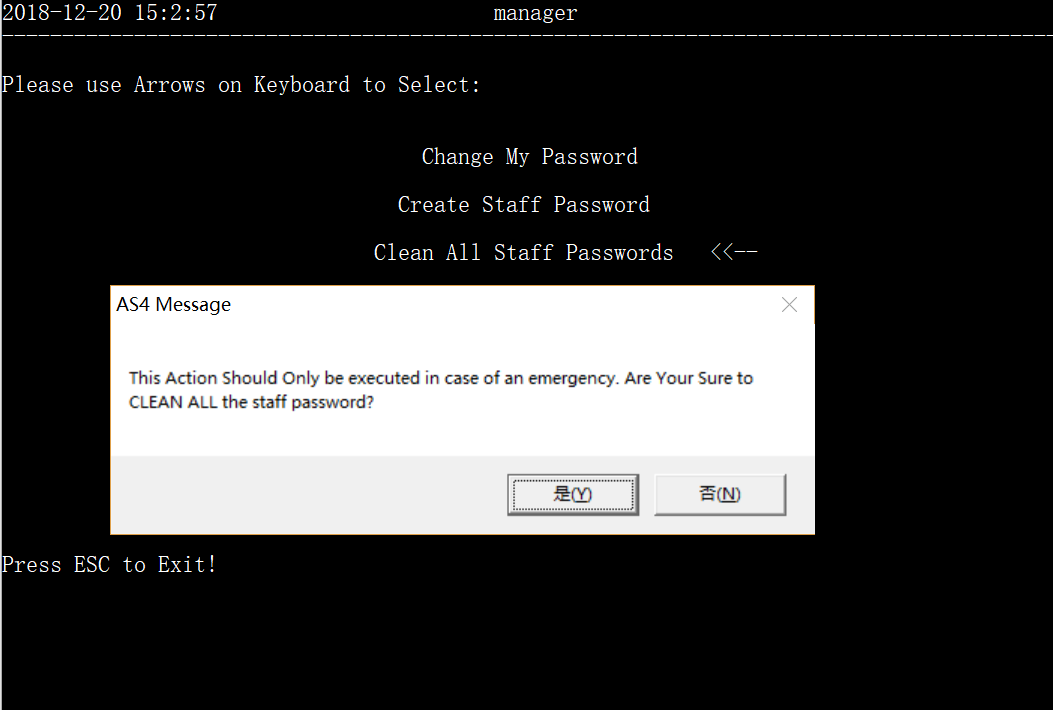
* Test 6: (after input the correct password of manager)



The program requires the manager to input the new password of the staff. Moreover, the procedures of doubling input of passwords and quitting the program after 5 times’ wrong input is the same as what is discussed before.

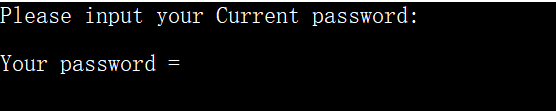
* Test 7: (clean all staff passwords)



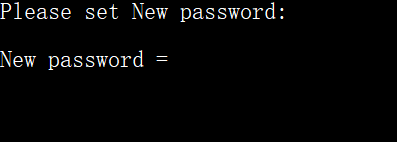


The program would to check and remind the manager that it WOULD CLEAN the staff’ passwords to prevent the accidental clicking of this button.

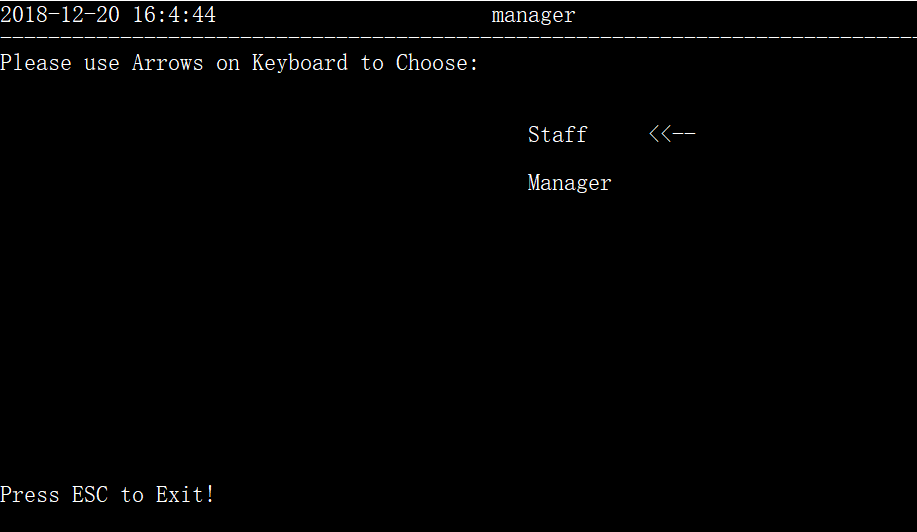
* Test 8: (pressing Ece while inputting the passwords)



When the program requires the manage input password when changing the passwords, the Ece button is pressed. The anticipated result is to go back to previous interface, but here, it directly skipping the procedures of inputting the passwords of manager, changing to the next step to setting the new passwords.



1. **Switch User**

****

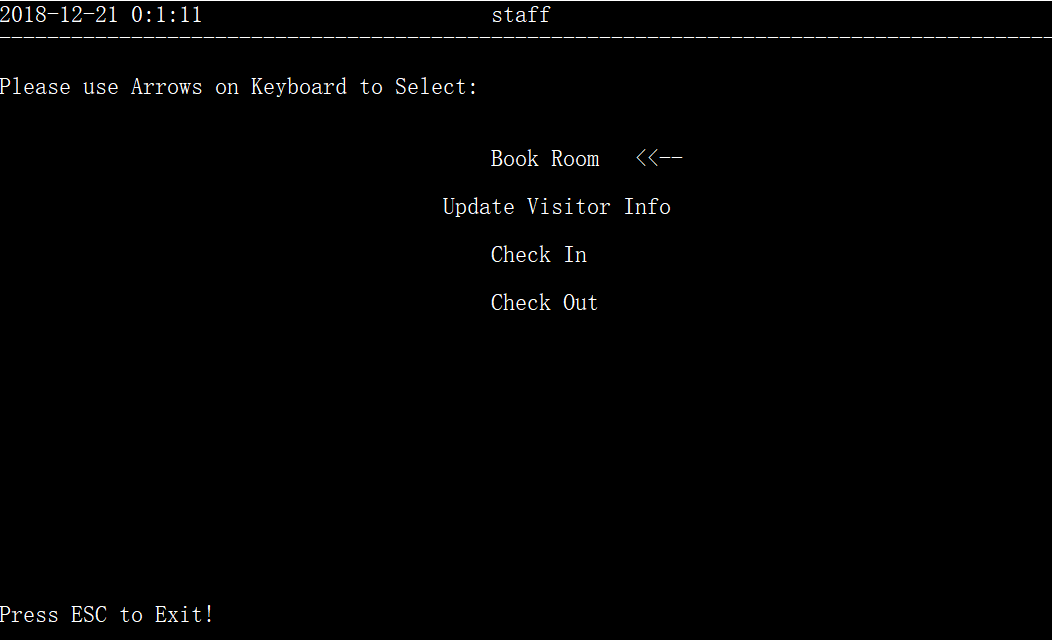
After selecting “switch user”, the program backs to the login interface as expected and the program loops to the first procedure.

1. **Exit**

****

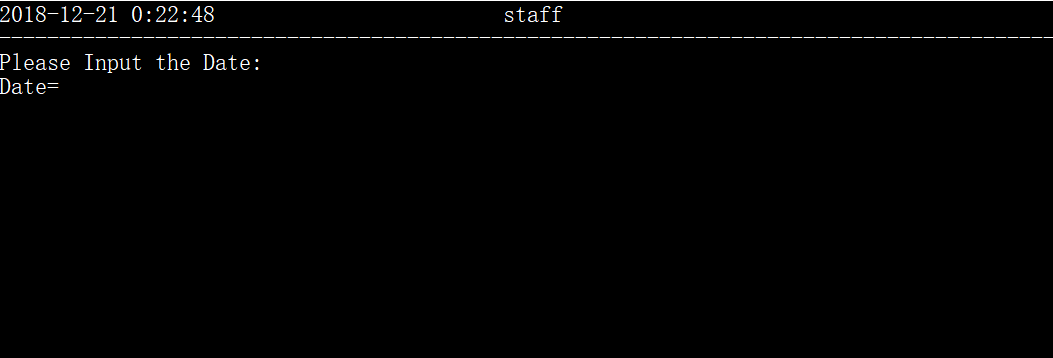
The program exits friendly with reminding messages displayed.

* 1. **Receptionist**

****

When the staff logs in, the next interface is printed as expected.

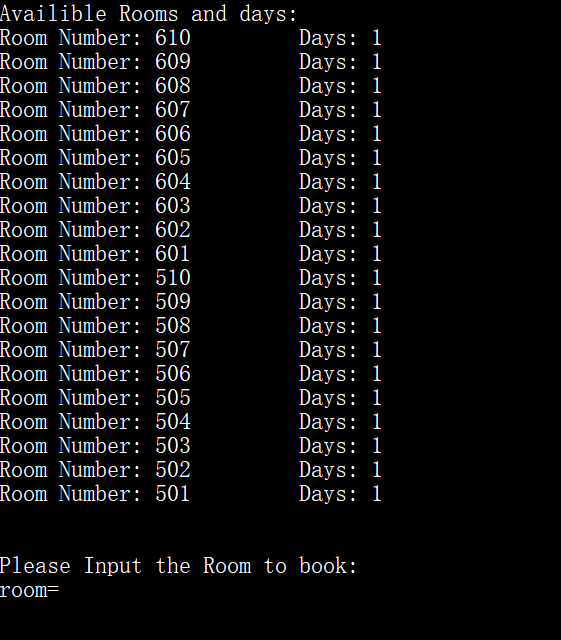
1. **Book Room**

****

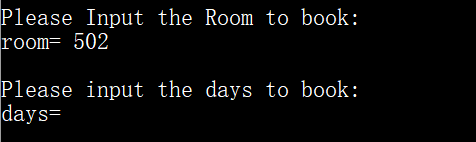
When entering “book room” option, the program expectedly requires the user to input the date in order to examine if there is room available on that day.

**(picture)**

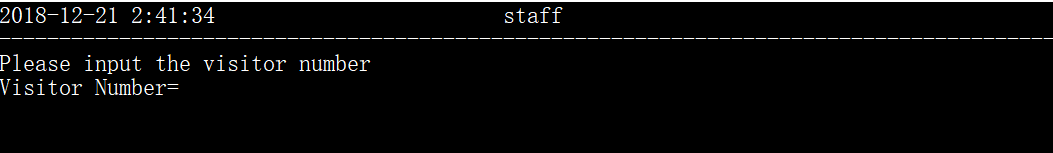
After inputting the date, the program asks user to input room type also in order to examine if there is room available.



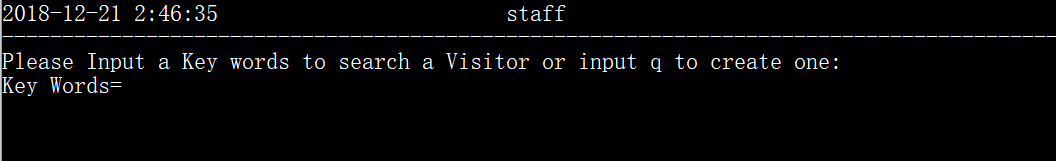
Then the program automatically searches and lists all the rooms that meet the requirements. It also marks the days that they can be booked. The program next asks the user to input the room needed to book. The procedures are all as expected.



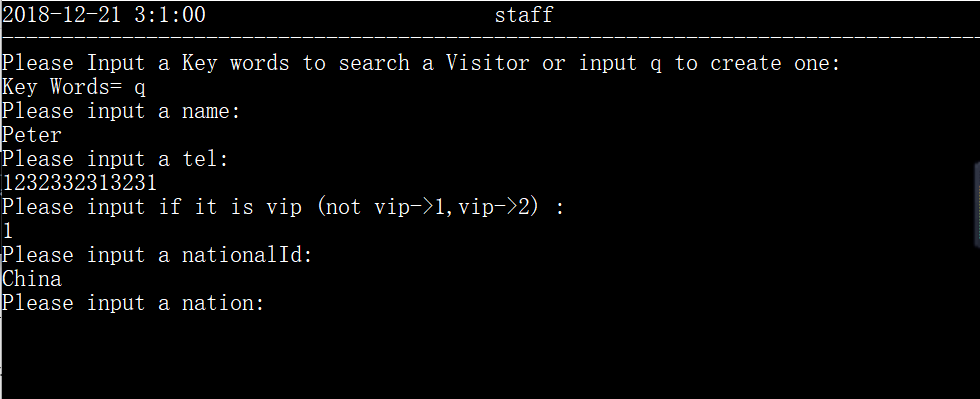
Then, the program expectedly asks the user to input room and days to book.



Then it asks to input the number of the visitors that need to book the room which is as expected



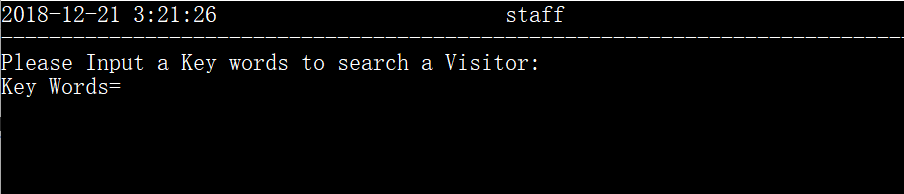
Next, it requires the user to enter key words for searching the identity of the user or create a new user identity.



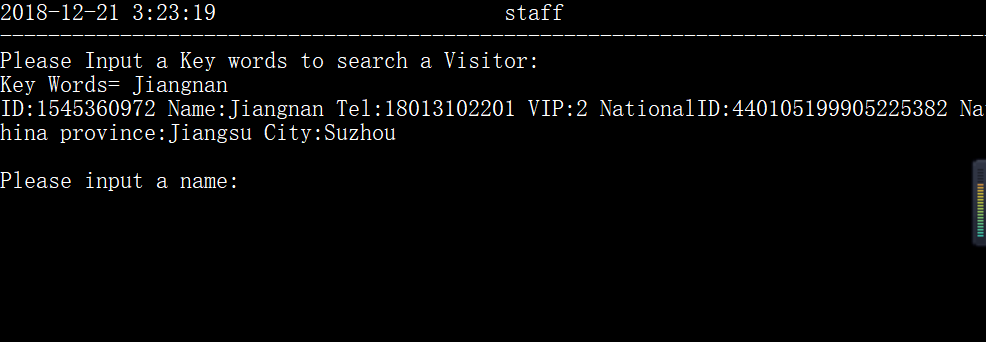
If selected creating a user, the program expectedly asks the user to input his/her basic information.

Then, the program automatically book the room for the user created which is as expected.

1. **Update visitor info**

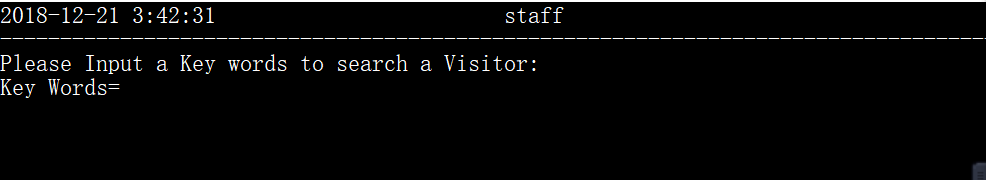
****

When entering the option, the program also requires the user to search the visitor so that it can edit the user’s information which is as expect.

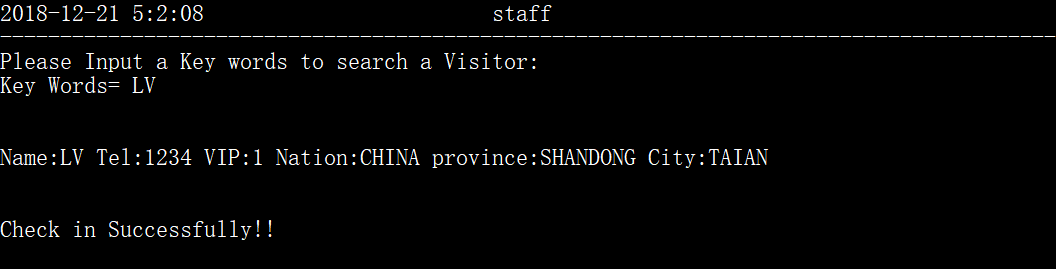


When the program finds the visitor’s information stored, it will expectedly require the user to input the new information and then finish editing.

**3）Check in**

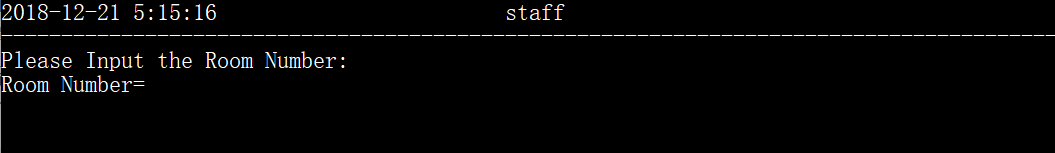
****

When enter the option, the program expectedly still requires the user to input a keyword to search the identity of the customer.

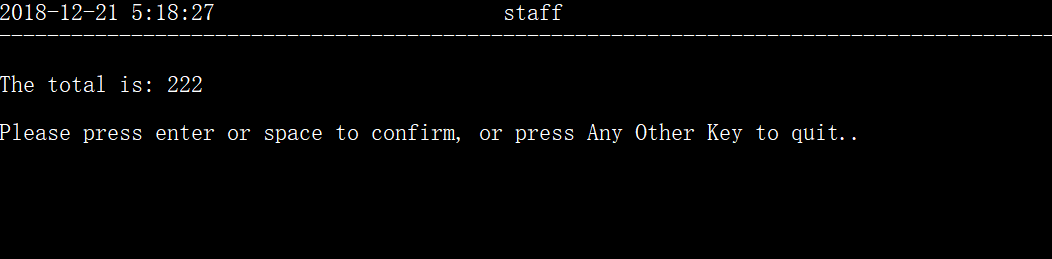


When it finds the information, the system will automatically check in and display reminding messages which is as expected. Next, the program will return to the last interface.

**4) Check out**



When enter the option, the user is required to enter the room number as expected.



Then the program displays the total cost which is as expected. After pressing enter to confirm, the program expectedly returns to the last interface.

During the whole testing process, the program runs as the expectation of the designed purpose.