Name : Tejeswara Sai Reddy Karri

Year : July – September 2021

Assignment : Create a Hotel Room Booking System in C Language

Table of Contents

INTRODUCTION

The prime goal of this project is to construct a computer program for Hotel Room Booking System for ease in managing workloads. The program code is written in C Language. The program is a simulation of hotel room booking for a week. The hotel has 2 floors and each floor with 5 rooms that is a total of 10 rooms. There are 5 types of rooms in this hotel. There is only one user for this program. It has options like View all rooms, Book a room, Search Booking details, Update booking details, View Receipt of Booked Room.

This task required the use of modular programming techniques, and I saved data in files and structures as needed. All entries are validated logically to avoid errors. To replicate this program in other computing languages, I created pseudocodes and flowcharts. The comprehensive C Language code with sample input and output images is provided below in this report. The program code is provided with meaningful comments to understand easily. I made a few assumptions before starting to write this program, which is listed in the next section.

ASSUMPTIONS

- This program is solely used by the admin of the Hotel and customers need to interact with admin for information.
- There is no payment page in this program as the payment is made to the admin manually.
- A customer can reserve only one room in a booking process.
- A service tax of 10% of total room charges and tourism tax of RM 10 per night is added to the bill.
- Customer who has booked a room is provided with a Booking ID which is important and must be noted by customer.
- A customer who has reserved a room can only change his Mobile Number, E-Mail, and Room details (which in turn changes his total charge).
- We have assumed the week to be in days i.e. (Monday to Sunday).

DESIGN OF THE PROGRAM

PSEUDOCODE

```
DECLARE STRUCTURE Customers.
DECLARE STRUCTURE room_master.
DECLARE FUNCTION Homepage
DECLARE FUNCTION disp_all_rooms
DECLARE FUNCTION rooms
DECLARE FUNCTION room_validation.
DECLARE FUNCTION search_bookings.
DECLARE FUNCTION receipts
DECLARE FUNCTION updates
DECLARE FUNCTION room_booking

CREATE a FUNCTION named as room_booking
    DECLARE variable customer from Customers(structure)
    DECLARE variable room from room_master(structure)
    DECLARE a STRING checkout_opt, checkin_opt
    DECLARE an INTEGER id

    DISPLAY a statement "----ROOM BOOKING----"
    DISPLAY a statement "----CHECK-IN DAY----"
    DISPLAY a statement (an option) "1. Monday"
    DISPLAY a statement (an option) "2. Tuesday"
    DISPLAY a statement (an option) "3. Wednesday"
    DISPLAY a statement (an option) "4. Thursday"
    DISPLAY a statement (an option) "5. Friday"
    DISPLAY a statement (an option) "6. Saturday"
    DISPLAY a statement (an option) "7. Sunday"
    DISPLAY a statement "Enter your option"
    Take USER INPUT into 'checkin_opt'
```

```
IF checkin_opt is in between 1 to 7 THEN
        continue
ELSE
        DISPLAY a statement "Invalid Input"
END IF

DISPLAY a statement "----CHECK-OUT DAY----"
DISPLAY a statement (an option) "1. Monday"
DISPLAY a statement (an option) "2. Tuesday"
DISPLAY a statement (an option) "3. Wednesday"
DISPLAY a statement (an option) "4. Thursday"
DISPLAY a statement (an option) "5. Friday"
DISPLAY a statement (an option) "6. Saturday"
DISPLAY a statement (an option) "7. Sunday"
DISPLAY a statement "Enter your option"
Take USER INPUT into 'checkout_opt'
IF checkout_opt is in between 1 to 7 THEN
        continue
ELSE
        DISPLAY a statement "Invalid Input"
END IF

CLEAR SCREEN
CONVERT checkin_opt and checkout_opt into integers and store them in check_in and check_out of customer structure respectively.
dos(days of stay) = check_out-check_in
DISPLAY check_in and check_out
IF dos(days of stay) is greater than 0 THEN
        DISPLAY dos
ELSE
        DISPLAY a statement "Invalid check in and check out dates. Try Again"\
        Exit
```

```
END IF
CALL room_validation FUNCTION
PROMPT user for his choice of room id and store it in room_id of customer structure

DECLARE a CHARACTER valid_room_id = 'N'
OPEN FILE available_rooms.txt in read mode
DOWHILE read each room details in available_rooms.txt iteratively
        IF room id of customer and room id of room are same THEN
                valid_room_id='Y'
        END IF
ENDWHILE
CLOSE FILE available_rooms.txt
IF valid_room_id=='N'
        Display a statement "Invalid Room ID entered. Please Try Again."
        Exit
ENDIF

PROMPT user for Name and store it in name of customer structure
PROMPT user for passport/IC Number and store it in psp_ic_num of customer structure
PROMPT user for Mobile Number and store it in cont_num of customer structure
DECLARE an INTEGER i=0, itisdigit=1
DOWHILE every index in cont_num string (cont_num[i])
        IF cont_num[i] is not a digit
                itisdigit=0
        ENDIF
        Increment in i by 1
ENDWHILE
IF length of cont_num string is less than 10 or greater than 11 OR itisdigit==0 THEN
        DISPLAY a statement "Invalid Mobile Number"
        Exit
ENDIF
```

```
    PROMPT user for E-Mail and store it in email of customer structure
    Generate a random value and store it in booking_id of customer structure

    OPEN FILE reserved_rooms_data.txt in append mode.
    WRITE the customer structure in reserved_rooms_data.txt file.
    CLOSE FILE reserved_rooms_data.txt
    DISPLAY a statement "Congrats!!! Your reservation has been confirmed !"
    DISPLAY the booking_id of customer structure
    DISPLAY a statement "(NOTE : Booking ID is important. Kindly Remember or note it down.)"
    CLEAR SCREEN
END OF FUNCTION room_booking

CREATE a FUNCTION named as updates
    DECLARE variable customer from Customers(structure)
    DECLARE variable room from room_master(structure)
    DECLARE a CHARACTER valid_booking_id = 'N', upd_opt
    DECLARE a STRING given_booking_id
    DECLARE an INTEGER size
    Assign the size of structure Customers to 'size' variable.
    DISPLAY a statement "-------UPDATE BOOKED ROOM-----"
    PROMPT user for his Booking ID and store in 'given_booking_id'

    OPEN FILE reserved_rooms_data.txt in read and write mode.
    DOWHILE read each reserved room details from reserved_rooms_data.txt file.
        IF given_booking_id and booking_id of customer are same THEN
            valid_booking_id = 'Y'
            DISPLAY a statement "What do you want to update ?"
            DISPLAY a statement (an option) "1. Contact Number"
            DISPLAY a statement (an option) "2. Email"
            DISPLAY a statement (an option) "3. Booking Details"
DISPLAY a statement (an option) "4. Exit"


DISPLAY a statement "Enter your option"
Take USER INPUT into 'upd_opt'


IF upd_opt = '1'
    PROMPT user to enter new contact number.
    Take USER INPUT into cont_num variable in customer
    structure.
    DECLARE an INTEGER i=0, itisdigit=1
    DOWHILE every index in cont_num string (cont_num[i])
        IF cont_num[i] is not a digit
            itisdigit=0
        ENDIF
        Increment in i by 1
    ENDWHILE
    IF length of cont_num string is less than 10 or greater than 11
    OR itisdigit==0 THEN
        DISPLAY a statement "Invalid Mobile Number"
        Exit
    ENDIF
    SEEK CURSOR to the beginning of the read customer in
    reserved_rooms_data.txt file.
    OVERWRITE the cusstomer with updated customer contact
    number.
    Exit
ELSE IF upd_opt = '2'
    PROMPT user to enter new E-Mail.
    Take USER INPUT into email variable in customer structure.
    SEEK CURSOR to the beginning of the read customer in
    reserved_rooms_data.txt file.
    OVERWRITE the cusstomer with updated customer E-Mail.
    Exit
ELSE IF upd_opt = '3'
```

```
DECLARE a STRING checkout_opt, checkin_opt
DISPLAY a statement "----UPDATING BOOKED ROOM----"
DISPLAY a statement "----UPDATING CHECK-IN DAY----"
DISPLAY a statement (an option) "1. Monday"
DISPLAY a statement (an option) "2. Tuesday"
DISPLAY a statement (an option) "3. Wednesday"
DISPLAY a statement (an option) "4. Thursday"
DISPLAY a statement (an option) "5. Friday"
DISPLAY a statement (an option) "6. Saturday"
DISPLAY a statement (an option) "7. Sunday"
DISPLAY a statement "Enter your option"
Take USER INPUT into 'checkin_opt'
IF checkin_opt is in between 1 to 7 THEN
        continue
ELSE
        DISPLAY a statement "Invalid Input"
        Exit
END IF

DISPLAY a statement "----UPDATING CHECK-OUT DAY----"
DISPLAY a statement (an option) "1. Monday"
DISPLAY a statement (an option) "2. Tuesday"
DISPLAY a statement (an option) "3. Wednesday"
DISPLAY a statement (an option) "4. Thursday"
DISPLAY a statement (an option) "5. Friday"
DISPLAY a statement (an option) "6. Saturday"
DISPLAY a statement (an option) "7. Sunday"
DISPLAY a statement "Enter your option"
Take USER INPUT into 'checkout_opt'
IF checkout_opt is in between 1 to 7 THEN
        continue
```

```
ELSE
    DISPLAY a statement "Invalid Input"
    Exit
END IF

CLEAR SCREEN
CONVERT checkin_opt and checkout_opt into integers and store them in check_in and check_out of customer structure respectively
dos(days of stay) = check_out-check_in
DISPLAY check_in and check_out
IF dos(days of stay) is greater than 0 THEN
    DISPLAY dos
ELSE
    DISPLAY a statement "Invalid check in and check out dates. Try Again"\
    Exit
END IF
CALL room_validation FUNCTION
PROMPT user for his choice of room id and store it in room_id of customer structure

DECLARE a CHARACTER valid_room_id = 'N'
OPEN FILE available_rooms.txt in read mode
DOWHILE read each room details in available_rooms.txt iteratively
IF room id of customer and room id of room are same THEN
        valid_room_id='Y'
    END IF
ENDWHILE
CLOSE FILE available_rooms.txt
IF valid_room_id=='N'
    Display a statement "Invalid Room ID entered. Please Try Again."
```

```
                        Exit
                    ENDIF
                    SEEK CURSOR to the beginning of the read customer in
                    reserved_rooms_data.txt file.
                    OVERWRITE the cusstomer with updated customer details.
                    Exit
                ELSE IF upd_opt = '4'
                    Exit
                ELSE
                    DISPLAY statement "Invalid input"
                END IF
            END IF
        ENDWHILE
        CLOSE FILE reserved_rooms_data.txt

        IF valid_booking_id == 'N' THEN
            DISPLAY a statement "ERROR : Invalid Booking ID entered. Please Try
            Again."
        END IF
        CLEAR SCREEN
    END OF FUNCTION updates
```

CREATE a FUNCTION named as receipts

```
        DECLARE variable room from room_master(structure)
        DECLARE variable customer from Customers(structure)
        DECLARE a CHARACTER valid_booking_id = 'N'
        DECLARE a STRING given_booking_id
        DECLARE an INTEGER total_charges

        DISPLAY a statement "-------RECEIPTS-----"
        PROMPT user for his Booking ID and store in 'given_booking_id'
        OPEN FILE reserved_rooms_data.txt in read mode.
        DOWHILE read each reserved room details from reserved_rooms_data.txt file.
        IF given_booking_id and booking_id of customer are same THEN
            valid_booking_id = 'Y'
            DISPLAY a statement "----YOUR BOOKING DETAILS----"
            DISPLAY name, passport/IC number, Contact Number, Email from
            the reserved room details
            DISPLAY a statement "---ROOM DETAILS---"
            DISPLAY Room ID, Check-in, and check-out date, Days of Stay from
            the reserved rooms details.

            OPEN FILE room_master.txt in read mode.
            DOWHILE read each room detail iteratively.
                IF room id of customer and room id of room are same THEN
                    DISPLAY the room type from room structure variable
                    total_charges = customer.dos * (10 + room.price * 11 /
                    10)
                    DISPLAY a statement "----TOTAL CHARGE------"
                    DISPLAY total_charges
                ENDIF
            ENDWHILE
            CLOSE FILE room_master.txt
        END IF
    ENDWHILE
    CLOSE FILE reserved_rooms_data.txt
    IF valid_booking_id == 'N' THEN
        DISPLAY a statement "ERROR : Invalid Booking ID entered. Please Try
        Again."
    END IF
    CLEAR SCREEN
END OF FUNCTION receipts

CREATE a FUNCTION named as search_bookings
    DECLARE variable customer from Customers(structure)
    DECLARE a CHARACTER valid_booking_id = 'N'
    DECLARE a STRING given_booking_id
    DISPLAY a statement "-------SEARCH BOOKING DETAILS-----"
    PROMPT user for his Booking ID and store it in 'given_booking_id'
    OPEN FILE reserved_rooms_data.txt in read mode.
    DOWHILE read each reserved room details from reserved_rooms_data.txt file till end
    of file.
        IF given_booking_id and booking_id of customer are same THEN
            valid_booking_id = 'Y'
            DISPLAY a statement "----YOUR BOOKING DETAILS----"
            DISPLAY name, passport/IC number, Contact Number, Email from
            the reserved room details
            DISPLAY a statement "---ROOM DETAILS---"
            DISPLAY Room ID, Check-in, and check-out date, Days of Stay from
            the reserved rooms details.
        END IF
    ENDWHILE
    CLOSE FILE reserved_rooms_data.txt
    IF valid_booking_id == 'N' THEN
        DISPLAY a statement "ERROR : Invalid Booking ID entered. Please Try
        Again."
    END IF
    CLEAR SCREEN
END OF FUNCTION search_bookings


CREATE a STRUCTURE named as Customers
    DECLARE INTEGER type variables check_in, check_out, dos(days of stay)
    DECLARE STRING type variables room_id, name, psp_ic_num, email, cont_num,
    booking_id
END OF STRUCTURE Customers


CREATE a STRUCTURE named as room_master
    DECLARE INTEGER type variables price
    DECLARE STRING type variables id, type.
END OF STRUCTURE room_master
```

```
CREATE a FUNCTION named as room_validation
    DECLARE variable room from room_master(structure)
    DECLARE variable customer from Customers(structure)
    DECLARE an INTEGER i
    DECLARE a CHARACTER is_room_available

    OPEN FILE room_master.txt in read mode.
    OPEN FILE available_rooms.txt in write mode.

    DISPLAY a statement "The below table represents the rooms available currently."
    DISPLAY a statement "ROOM ID--ROOM TYPE--ROOM PRICE PER DAY"
    DOWHILE read each room details in room_master.txt file till end of file
        is_room_available = 'Y'
        OPEN FILE reserved_rooms_data.txt in read mode.
        DOWHILE read each reserved room details from reserved_rooms_data.txt
        file.
            IF room id of customer and room id of room are same THEN
                FOR i = ck_in and i < ck_out + 1 STEP 1
                    IF i >= customer.check_in AND i < customer.check_out
                    THEN
                        is_room_available = 'N'
                        BREAK
                    END IF
                ENDFOR
            ENDIF
        END WHILE
        CLOSE FILE reserved_rooms_data.txt
        IF is_room_available == 'N' THEN
            CONTINUE to next iteration.
        END IF
        DISPLAY  the room.id , room.type and room.price
        WRITE the room details to available_rooms.txt file.
    END WHILE
    CLOSE FILE available_rooms.txt
    CLOSE FILE room_master.txt
END OF FUNCTION room_validation

CREATE a FUNCTION named as rooms
    Assign the values to room_master structure room1 = {"101","Superior",180}
    Assign the values to room_master structure room2 = {"102","Deluxe",200}
    Assign the values to room_master structure room3 = {"103","Studio",250}
    Assign the values to room_master structure room4 = {"104","Executive Suite",400}
    Assign the values to room_master structure room5 = {"105","Deluxe Suite",500}
    Assign the values to room_master structure room6 = {"201","Superior",180}
    Assign the values to room_master structure room7 = {"202","Deluxe",200}
    Assign the values to room_master structure room8 = {"203","Studio",250}
    Assign the values to room_master structure room9 = {"204","Executive Suite",400}
    Assign the values to room_master structure room10 = {"205","Deluxe Suite",500}

    OPEN FILE room_master.txt in read mode.

    WRITE room1 in room_master.txt file.
    WRITE room2 in room_master.txt file.
    WRITE room3 in room_master.txt file.
    WRITE room4 in room_master.txt file.
    WRITE room5 in room_master.txt file.
    WRITE room6 in room_master.txt file.
    WRITE room7 in room_master.txt file.
    WRITE room8 in room_master.txt file.
    WRITE room9 in room_master.txt file.
    WRITE room10 in room_master.txt file.

    CLOSE FILE room_master.txt.
END OF FUNCTION rooms.
```

```
CREATE a FUNCTION named as disp_all_rooms
      OPEN FILE room_master.txt in read mode.
      DISPLAY a statement "---ALL ROOMS---" as Title
      DISPLAY a statement "ROOM ID--ROOM TYPE--ROOM PRICE PER DAY".
      DOWHILE read a room details (structure).
            DISPLAY the room details.
      ENDWHILE.
      CLOSE FILE room_master.
      DISPLAY a statement "You can book the rooms by selecting option '2' in
      HOMEPAGE"
      CLEAR SCREEN.
END OF FUNCTION disp_all_rooms

CREATE a FUNCTION named as Homepage
  DECLARE opt(Option) as a string variable and prompt_homepage as int variable
  DOWHILE loop with condition prompt_homepage = 1
    Print the statement "Hotel Room Booking System" as Title
    DISPLAY a statement (an option) "1. See available rooms"
    DISPLAY a statement (an option) "2. Book Room"
    DISPLAY a statement (an option) "3. Search Booking details"
    DISPLAY a statement (an option) "4. Update Booking details"
    DISPLAY a statement (an option) "5. View Receipt"
    DISPLAY a statement (an option) "6. Exit"

    DISPLAY a statement "Enter your option"
    Take USER INPUT into 'opt'

    IF opt = '1'
      CALL disp_all_rooms FUNCTION
    ELSE IF opt = '2'
      CALL room_booking FUNCTION
```

```
        ELSE IF opt = '3'
            CALL search_bookings FUNCTION
        ELSE IF opt = '4'
            CALL updates FUNCTION
        ELSE IF opt = '5'
            CALL receipts FUNCTION
        ELSE IF opt = '6'
            DISPLAY statement "Thank You ..."
        ELSE
            DISPLAY statement "Invalid input"
        END IF
    END WHILE
END OF FUNCTION Homepage

CALL HOMEPAGE FUNCTION
```

FLOWCHARTS

Note :- The Flow charts for this assignment are available in the below OneDrive Link. Kindly access the link to see the flowcharts in a Visio File.

Link :-

https://cloudmails-my.sharepoint.com/:u:/g/personal/tp062689_mail_apu_edu_my/EWOk7rnfpVhNqzzYfoRUWv4BtOiXyvAD4FvM1hvZqnWZxA

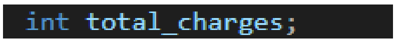
EXPLANATION OF C CONCEPTS USED

1. Data Types

Data types are the format in which data is stored in the system. There are mainly three data type namely Integer, Float, Character and Arrays. Arrays are compound data types that store multiple elements of the same data type. There can be arrays of Integer, Character and Float data types.

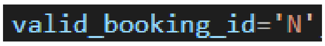
I have only used Integer and Character Data types in this assignment, a few are given below.

Integer – Stores data only of integers



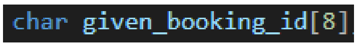
```
int total_charges;
```

Character – Stores data only of a single character.



```
valid_booking_id='N'
```

Character Array – Stores multiple character data types.

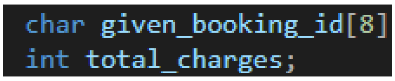


```
char given_booking_id[8]
```

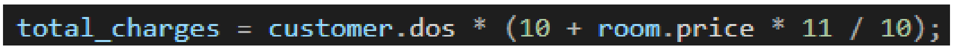
It is represented by adding Brackets to the variable with the number of elements in it as per above figure.

2. Variables

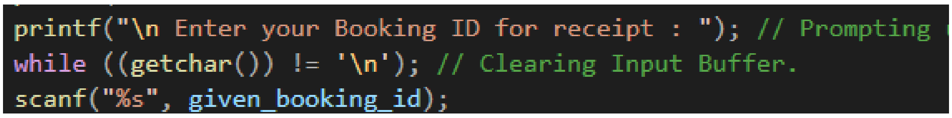
Variables are temporary storage locations to store data in memory. When u define a variable, the system allocates a location in memory addressing it as the variable name. Later when u assign a value to the variable it is stored at a location in memory temporarily.



```
char given_booking_id[8]
int total_charges;
```



```
total_charges = customer.dos * (10 + room.price * 11 / 10);
```

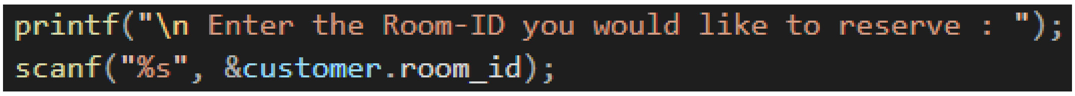


```
printf("\n Enter your Booking ID for receipt : "); // Prompting
while ((getchar()) != '\n'); // Clearing Input Buffer.
scanf("%s", given_booking_id);
```

I defined variables given_booking_id (char array type) and total_charges (int type). The total charge of the customers booking is calculated and assigned to total_charges variable and stored temporarily as it changes for different customers.

3. Input / Output

Output is the way of displaying data on screen. Input is receiving data from user. It can be performed using scanf and printf statements.



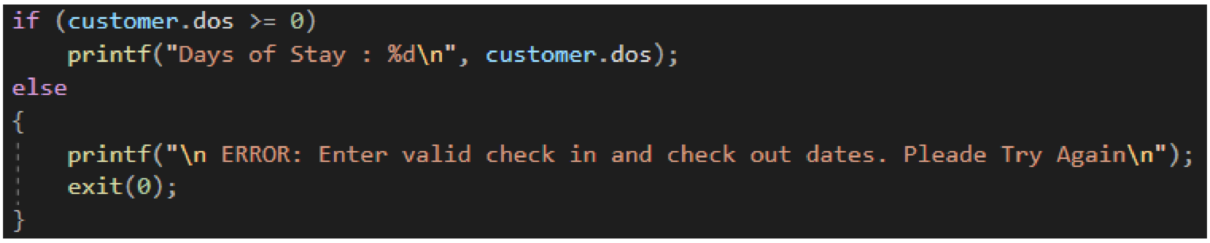
```
printf("\n Enter the Room-ID you would like to reserve : ");
scanf("%s", &customer.room_id);
```

As per the above image, the program prints the Statement in printf and receives whatever user types into customer.room_id .

4. Control Structures

If – Else

An if – else loop is a simple condition in which a particular statement or process is executed depending on a condition. If a condition is satisfied one route is taken else another route is taken.



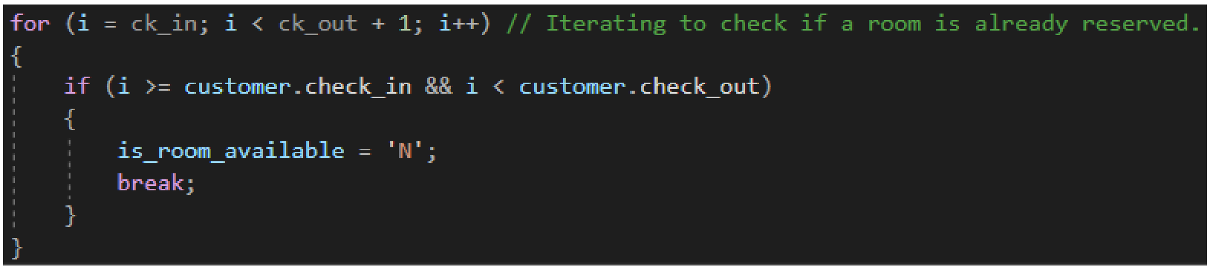
```
if (customer.dos >= 0)
    printf("Days of Stay : %d\n", customer.dos);
else
{
    printf("\n ERROR: Enter valid check in and check out dates. Pleade Try Again\n");
    exit(0);
}
```

In the above pic, u can see that only if customer.dos(variable) is greater than 0 then Days of Stay is printed else it prints the error message and exits the program.

Loops

Loops are simple conditions which repeat defined statements as per conditions. There are two types of loops For and While Loops. Both perform the same function with different syntax.

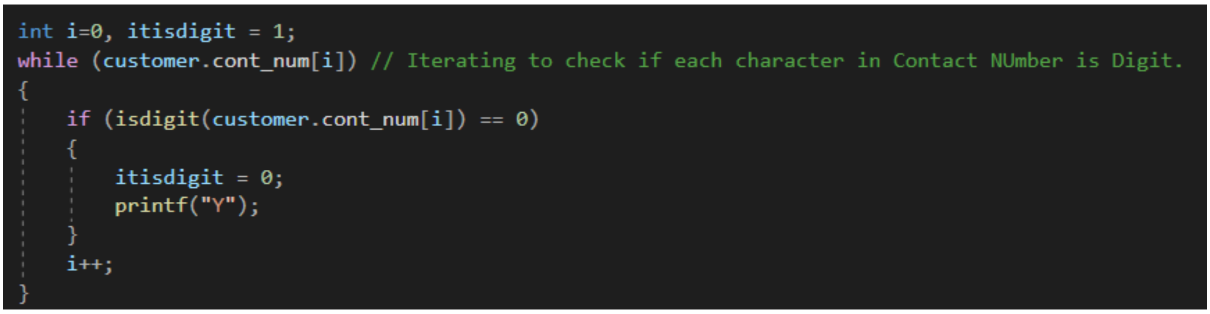
For Loop :-



```
for (i = ck_in; i < ck_out + 1; i++) // Iterating to check if a room is already reserved.
{
    if (i >= customer.check_in && i < customer.check_out)
    {
        is_room_available = 'N';
        break;
    }
}
```

As per the above image, the loop runs an if statement based on the condition. The for loop here has an initiator (i = ck_in) , if i is less than ck_out + 1 then the if statement runs. The value of i is increased by 1 in every loop, which is written as i++, it is also called as Increment. It runs repeatedly till the condition is false.

While Loop :-

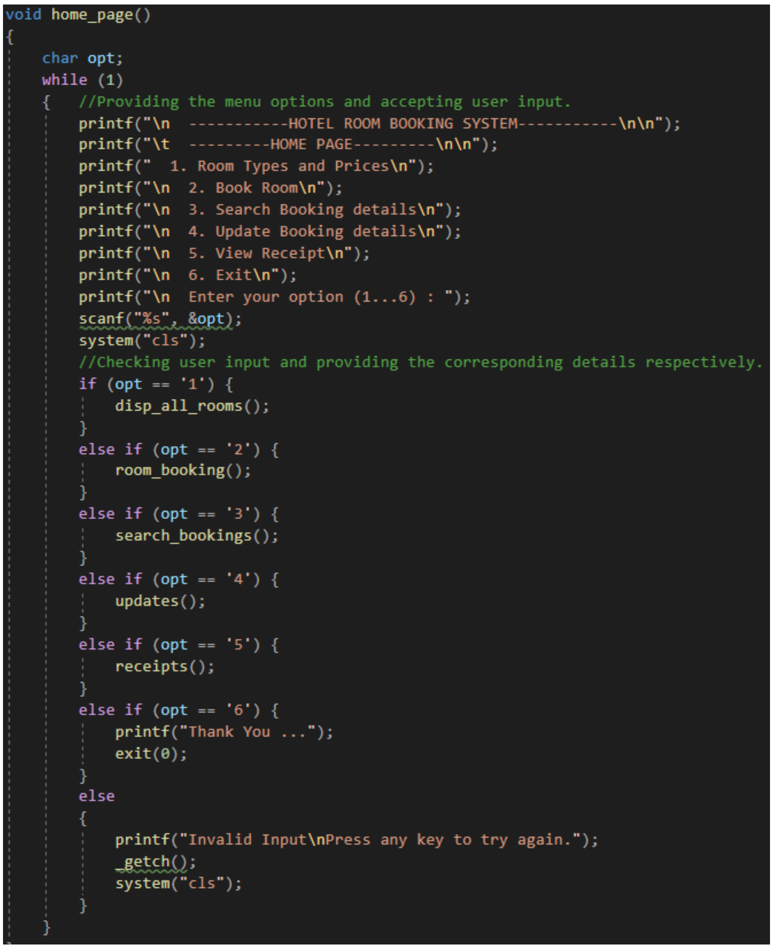


```
int i=0, itisdigit = 1;
while (customer.cont_num[i]) // Iterating to check if each character in Contact NUmber is Digit.
{
    if (isdigit(customer.cont_num[i]) == 0)
    {
        itisdigit = 0;
        printf("Y");
    }
    i++;
}
```

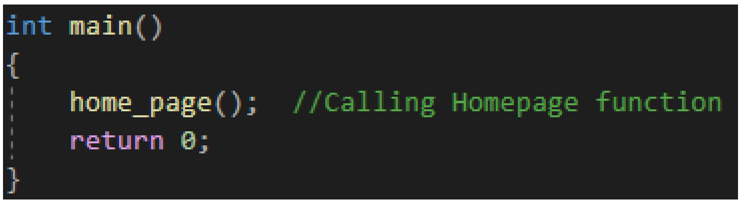
As per the above image, the while loop runs an if statement till the variable customer.cont_num[i] has a value which returns true. Unlike the For loop the initiator and the increment is not mentioned in the condition. The initiator (i=0) is defined before the syntax of while loop and the increment is provided in the statements block of while loop at the end.

5. Functions

Functions are codes that accomplish a specific task. A function can be defined and called multiple times in the code space. There are built-in and user defined functions. Built-in functions are printf(), scanf(), isdigit(), etc. which are predefined in the compiler by the language. User defined functions are created by users for various purposes.



```c
void home_page()
{
    char opt;
    while (1)
    {   //Providing the menu options and accepting user input.
        printf("\n  -----------HOTEL ROOM BOOKING SYSTEM-----------\n\n");
        printf("\t  ---------HOME PAGE---------\n\n");
        printf("  1. Room Types and Prices\n");
        printf("\n  2. Book Room\n");
        printf("\n  3. Search Booking details\n");
        printf("\n  4. Update Booking details\n");
        printf("\n  5. View Receipt\n");
        printf("\n  6. Exit\n");
        printf("\n  Enter your option (1...6) : ");
        scanf("%s", &opt);
        system("cls");
        //Checking user input and providing the corresponding details respectively.
        if (opt == '1') {
            disp_all_rooms();
        }
        else if (opt == '2') {
            room_booking();
        }
        else if (opt == '3') {
            search_bookings();
        }
        else if (opt == '4') {
            updates();
        }
        else if (opt == '5') {
            receipts();
        }
        else if (opt == '6') {
            printf("Thank You ...");
            exit(0);
        }
        else
        {
            printf("Invalid Input\nPress any key to try again.");
            _getch();
            system("cls");
        }
    }
```

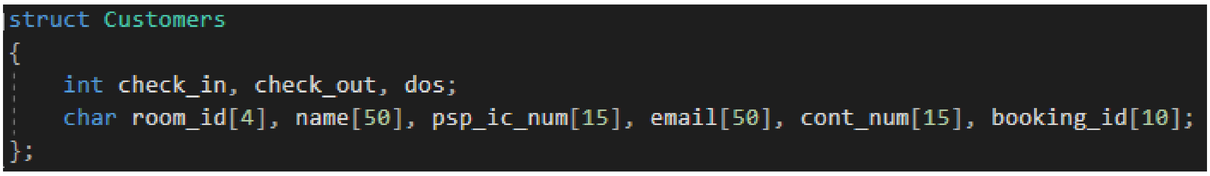


```c
int main()
{
    home_page();  //Calling Homepage function
    return 0;
}
```

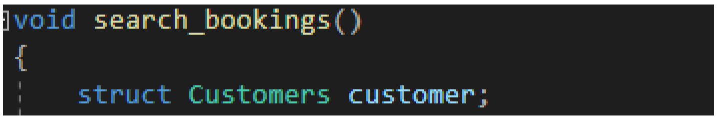
The above image is a function called homepage and is defined as shown i.e., when homepage is called it runs the code in it.

6. Structures

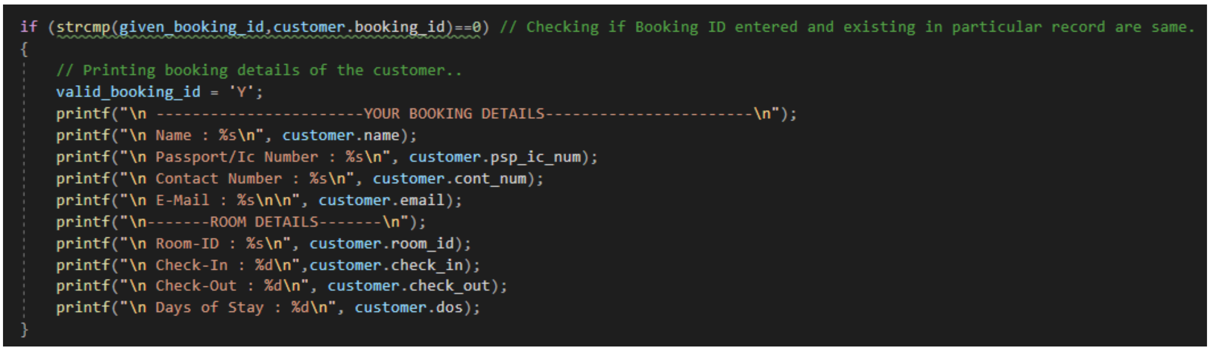
Structure is a compound data type that can store multiple elements of same or different data type. Structures are mainly used to store various details about the same thing or person.



```c
struct Customers
{
    int check_in, check_out, dos;
    char room_id[4], name[50], psp_ic_num[15], email[50], cont_num[15], booking_id[10];
};
```



```c
void search_bookings()
{
    struct Customers customer;
```

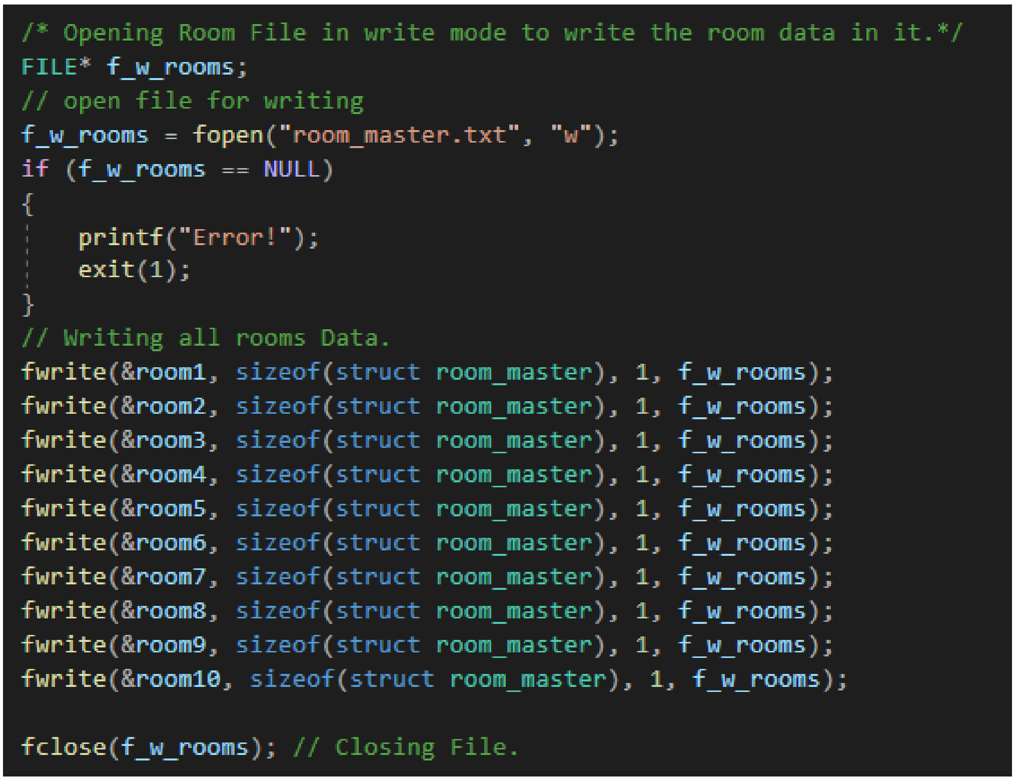


```c
if (strcmp(given_booking_id,customer.booking_id)==0) // Checking if Booking ID entered and existing in particular record are same.
{
    // Printing booking details of the customer..
    valid_booking_id = 'Y';
    printf("\n ----------------------YOUR BOOKING DETAILS----------------------\n");
    printf("\n Name : %s\n", customer.name);
    printf("\n Passport/Ic Number : %s\n", customer.psp_ic_num);
    printf("\n Contact Number : %s\n", customer.cont_num);
    printf("\n E-Mail : %s\n\n", customer.email);
    printf("\n-------ROOM DETAILS-------\n");
    printf("\n Room-ID : %s\n", customer.room_id);
    printf("\n Check-In : %d\n",customer.check_in);
    printf("\n Check-Out : %d\n", customer.check_out);
    printf("\n Days of Stay : %d\n", customer.dos);
}
```

The above image is a structure defined as Customers has integer and character array type variables. It is called to use the variables wherever necessary. For example, it is called in search booking function to compare the customer booking id and the booking id entered.

7. Files and Pointers

Files are storage locations outside the compiler and stored in permanent storage of memory. Data stored in files remains the same even when compilation ends. Pointers are special variables that stores variable address. These are mainly used for a different method to store data.



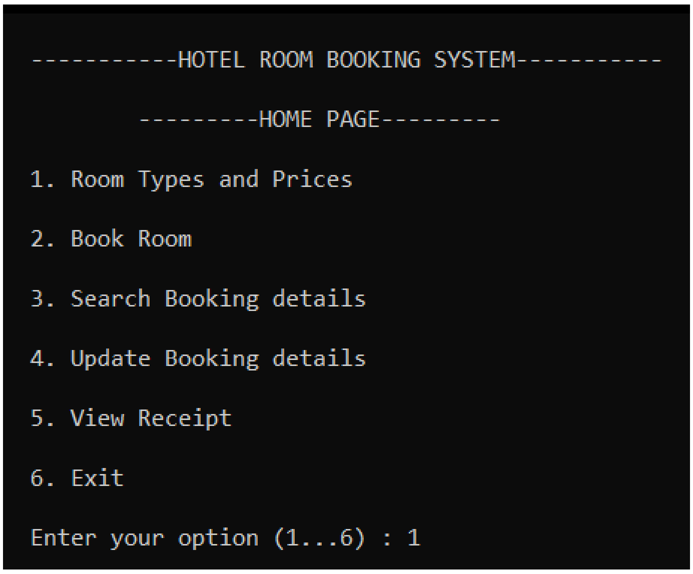
```
/* Opening Room File in write mode to write the room data in it.*/
FILE* f_w_rooms;
// open file for writing
f_w_rooms = fopen("room_master.txt", "w");
if (f_w_rooms == NULL)
{
    printf("Error!");
    exit(1);
}
// Writing all rooms Data.
fwrite(&room1, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room2, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room3, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room4, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room5, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room6, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room7, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room8, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room9, sizeof(struct room_master), 1, f_w_rooms);
fwrite(&room10, sizeof(struct room_master), 1, f_w_rooms);

fclose(f_w_rooms); // Closing File.
```

From the above image, pointers are used to access a file. f_w_rooms is a file pointer which points to a memory address that stores info in a file temporarily in many modes. In our case it is stored in write mode, to write data into the file. After the data is written in the pointer it is closed, at this point the data in pointers is taken back to the file and stored permanently.

SAMPLE INPUT/OUTPUT

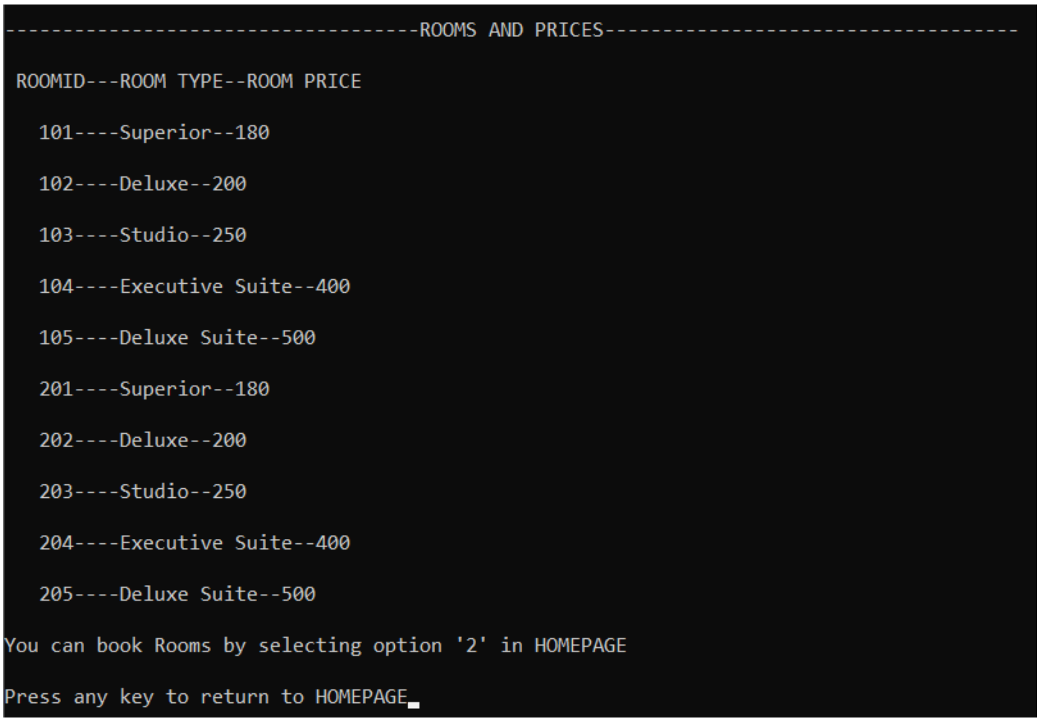
1. To View available room and prices.

Input



```
-----------HOTEL ROOM BOOKING SYSTEM-----------

        ---------HOME PAGE---------

1. Room Types and Prices

2. Book Room

3. Search Booking details

4. Update Booking details

5. View Receipt

6. Exit

Enter your option (1...6) : 1
```

Output

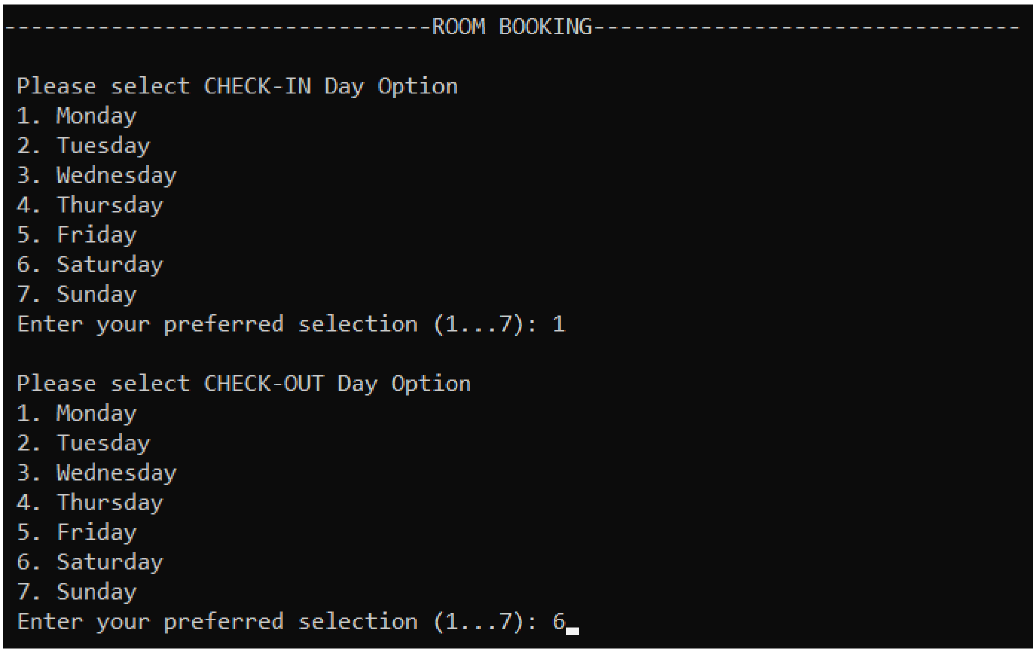


```
-----------------------------------ROOMS AND PRICES-----------------------------------

 ROOMID---ROOM TYPE--ROOM PRICE

   101----Superior--180

   102----Deluxe--200

   103----Studio--250

   104----Executive Suite--400

   105----Deluxe Suite--500

   201----Superior--180

   202----Deluxe--200

   203----Studio--250

   204----Executive Suite--400

   205----Deluxe Suite--500

You can book Rooms by selecting option '2' in HOMEPAGE

Press any key to return to HOMEPAGE_
```

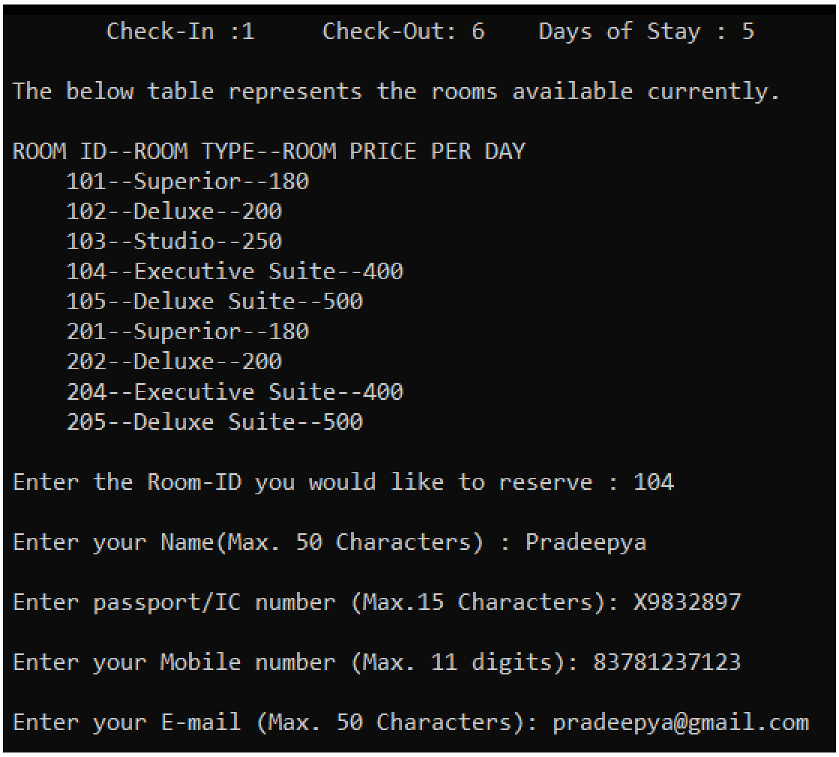
If the user enters 1, it will provide an output about the Available rooms and prices.

2. Book a Room

Input



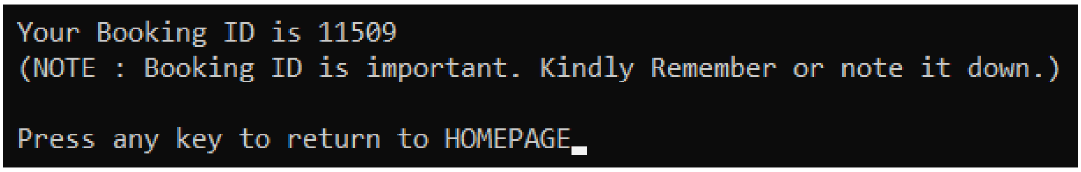
```
--------------------------------ROOM BOOKING--------------------------------

 Please select CHECK-IN Day Option
 1. Monday
 2. Tuesday
 3. Wednesday
 4. Thursday
 5. Friday
 6. Saturday
 7. Sunday
 Enter your preferred selection (1...7): 1

 Please select CHECK-OUT Day Option
 1. Monday
 2. Tuesday
 3. Wednesday
 4. Thursday
 5. Friday
 6. Saturday
 7. Sunday
 Enter your preferred selection (1...7): 6
```



```
        Check-In :1      Check-Out: 6     Days of Stay : 5

 The below table represents the rooms available currently.

 ROOM ID--ROOM TYPE--ROOM PRICE PER DAY
     101--Superior--180
     102--Deluxe--200
     103--Studio--250
     104--Executive Suite--400
     105--Deluxe Suite--500
     201--Superior--180
     202--Deluxe--200
     204--Executive Suite--400
     205--Deluxe Suite--500

 Enter the Room-ID you would like to reserve : 104

 Enter your Name(Max. 50 Characters) : Pradeepya

 Enter passport/IC number (Max.15 Characters): X9832897

 Enter your Mobile number (Max. 11 digits): 83781237123

 Enter your E-mail (Max. 50 Characters): pradeepya@gmail.com
```

When user inputs the check-in and check-out day, program clears screen and displays selected check-in, check-out date, Days of stay calculated by the program current available rooms and prompts for choice of room with Room ID, then user enters his personal details like name, email, contact Number and Passport Number.

Output

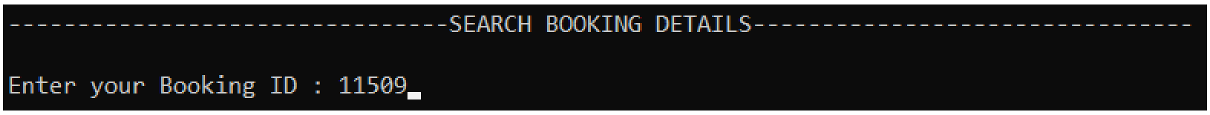


```
Your Booking ID is 11509
(NOTE : Booking ID is important. Kindly Remember or note it down.)

Press any key to return to HOMEPAGE
```

When user inputs his data and hits enter, if all data inputted is valid, then the program generates a Booking ID and outputs it.

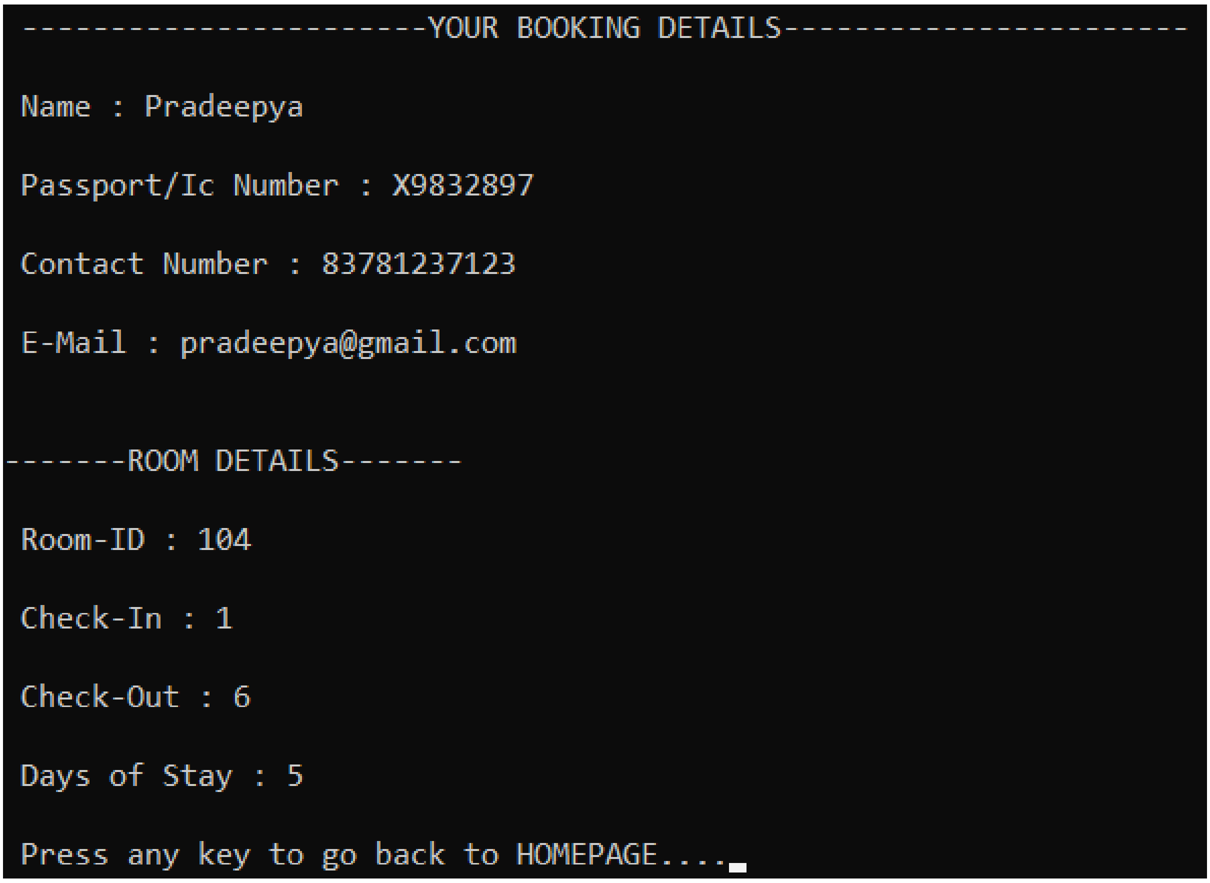
3. Search Booking Details

Input



```
------------------------------SEARCH BOOKING DETAILS------------------------------

Enter your Booking ID : 11509
```

Output

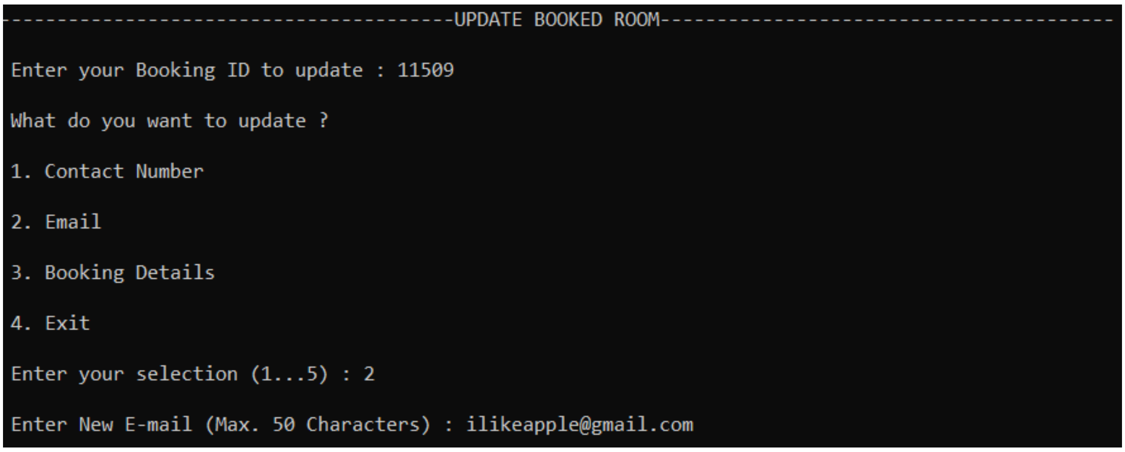


```
 ----------------------YOUR BOOKING DETAILS----------------------

 Name : Pradeepya

 Passport/Ic Number : X9832897

 Contact Number : 83781237123

 E-Mail : pradeepya@gmail.com


-------ROOM DETAILS-------

 Room-ID : 104

 Check-In : 1

 Check-Out : 6

 Days of Stay : 5

 Press any key to go back to HOMEPAGE....
```

When user enters his Booking ID, provided to him after room booking, his booking details is displayed.

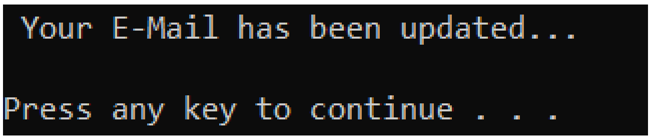
4. Update Booking Details

Input



```
-------------------------------------UPDATE BOOKED ROOM-------------------------------------

Enter your Booking ID to update : 11509

What do you want to update ?

1. Contact Number

2. Email

3. Booking Details

4. Exit

Enter your selection (1...5) : 2

Enter New E-mail (Max. 50 Characters) : ilikeapple@gmail.com
```

Output

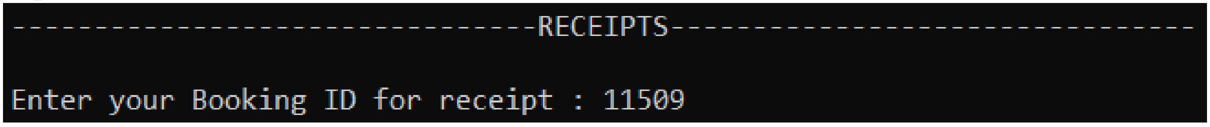


```
 Your E-Mail has been updated...

Press any key to continue . . .
```

When user enters his Booking ID, it prompts to choose his choice of details to be updated, user selects it and updates it, then it notifies it has been updated.

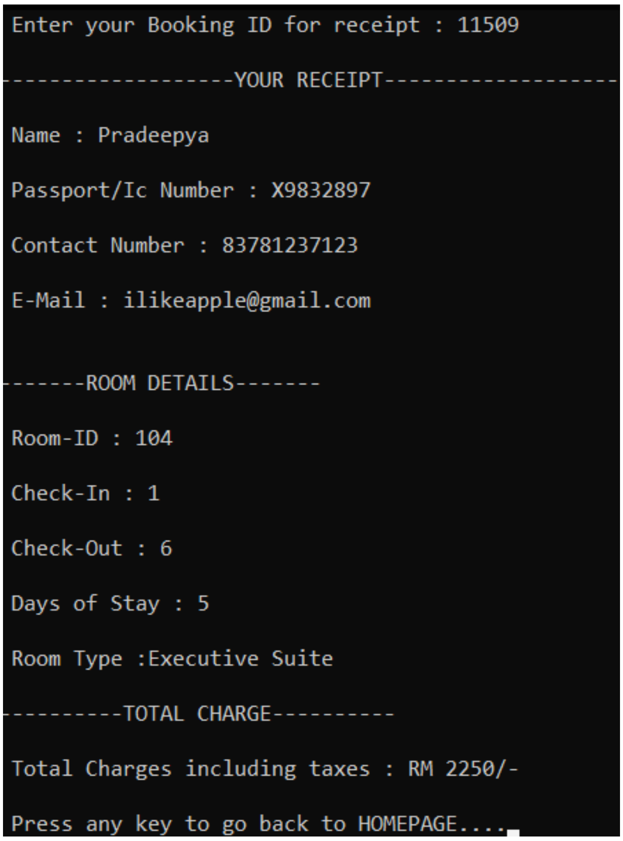
5. Receipt

Input



```
--------------------------------RECEIPTS--------------------------------

Enter your Booking ID for receipt : 11509
```

Output



```
Enter your Booking ID for receipt : 11509

------------------YOUR RECEIPT------------------

Name : Pradeepya

Passport/Ic Number : X9832897

Contact Number : 83781237123

E-Mail : ilikeapple@gmail.com


------ROOM DETAILS-------

Room-ID : 104

Check-In : 1

Check-Out : 6

Days of Stay : 5

Room Type :Executive Suite

---------TOTAL CHARGE----------

Total Charges including taxes : RM 2250/-

Press any key to go back to HOMEPAGE....
```

When user enters his Booking ID provided, his receipt is displayed that includes his booking details and the total charge calculated with taxes.

CONCLUSION

To conclude the given assignment Hotel Room Booking System, all requirements of the system have been met. The program design (Pseudocode and flowchart) is created and implemented in the program. All logical assumption mentioned have been used in the program. Proper naming conventions, comments, variables, files, structures, functions are used extensively. All features of the program View all rooms, book a room, Search Booking details, Update booking details, View Receipt of Booked Room are available in the program. The submitted program is fit for adding UI and real-life applications.