

Yaakoub Khalil 12200055

XCM

AMERICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CSI 205: COMPUTER PROGRAMMING I Fall Term 2021-2022 FINAL PROJECT

Table of Contents

**Problem 12**

Problem definition2

Commented source code………………………………………………………………………………………………………………………………………….………2

Discussion about the program……………………………………………………………………………………………………………………………………….4

Program output……………………………………………………………………………………………………………………………………………………….…..5

Soft copy of the source code5

**Problem 26**

Problem definition 6

Commented source code7

Discussion about the program………………………………………………………………………………………………………..………………………………14

Discussion about the function……………………………………………………………………………………………………………………….………...…14

Program output……………………………………………………………………………………………………………………………………………………….…15

Soft copy of the source code……………………………………………………………………………………………………………………………….…….15

**Problem 3………………………………………………………………………………………………………………………………………………….……………….16**

Problem definition………………………………………………………………………………………………………………………………………………………….16

Commented source code……………………………………………………………………………………………………………………………………………….17

Discussion about the program………………………………………………………….………………………………………………………………….………24

Discussion about the function…………………………………………………………………………………………………………………………………….24

Program output……………………………………………………………………………………………………………………………………………………….25

Soft copy of the source code……………………………………………………………………………………………………………….……………....26

**Problem 1:**

The Global\_com telephone company has the following rate structure for calls:

1. The regular rate for a call is $0.40 per minute.

2. Any call started after 6:00 P.M. (18 on 24-hour clock) but before 8:00 A.M. (8 on 24-

hour clock) is discounted 50%.

3. All calls are subject to 4% federal tax.

4. Any call longer than 60 minutes receives a 15% discount on its cost.

Write a program to calculate the telephone bill of a client which is calculated by adding up

the net cost of all telephone calls on the bill.

First the program should prompt the user for the number of calls. Then for each telephone call

the user should input the start time for the call based on 24-hour clock, the end time for the

call based on 24-hour clock.

The program should output the gross cost (before any discounts or tax) followed by the net

cost (after discounts are deduced and tax is added) for each call.

At the end the program should output the value of the client bill

**PROGRAM 1**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int calls; // number of phone calls done by the user.

double minuteCharge = 0.4; // charge of the phone call(1 minute).

int startHour = 0, startMinute = 0; // this means the hour and the minut the call starts at.

int endHour = 0, endMinute = 0; // here means the hour and the minut the call ends at.

int callDuration = 0; // how much time the user talk on the phone.

double callCharge = 0.0;

double tax = 1.04;

double grossCost = 0.0; // before discount and before tax.

double netCost = 0.0; // after discount and after tax.

cout << "Enter the number of phone calls: " << endl;

cin >> calls;

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

{

cout << "Enter the time when the user calls the number " << i + 1 << " started: " << endl;

cout << "Start time hour(24-hour clock): ";// the time in hours, for example 16:30, 16 is the start time hours.

cin >> startHour;

cout << "Start time minutes: "; // the time in minutes, for example 16:30, 30 is the start time minutes.

cin >> startMinute;

cout << endl;

cout << "Enter the time when call number " << i + 1 << " ended: " << endl;

cout << "End time hour(24-hour clock): "; // same as start time hour.

cin >> endHour;

cout << "End time minutes: "; // same as start time minute.

cin >> endMinute;

cout << endl;

callDuration = abs(endHour - startHour) \* 60 + abs(endMinute - startMinute); // convert to min

callCharge = callDuration \* minuteCharge; // charge of the call

grossCost += callCharge;

cout << "\_\_\_\_\_" << endl;

cout << "Information about call number " << i + 1 << ":" << endl;

cout << "-----------------------------------------" << endl;

cout << "Total minutes: " << callDuration << endl;

cout << "Cost before any discounts: " << callCharge << "$" << endl;

if (callDuration > 60)

{

callCharge = callCharge \* 0.85; // charge of the call + 15% discount

}

if ((startHour >= 18 || startHour <= 8) && (endHour >= 18 || endHour <= 8))

{

callCharge = callCharge / 2; // 50% discount

}

cout << "Cost after discounts and tax: " << callCharge \* tax << "$" << endl;

cout << "\_\_\_\_\_" << endl << endl;

netCost += callCharge \* tax;

}

cout << "Total Gross Cost is: " << grossCost << "$" << endl;

cout << "Federal tax = 4%" << endl;

cout << "Total bill is: " << netCost << "$" << endl;

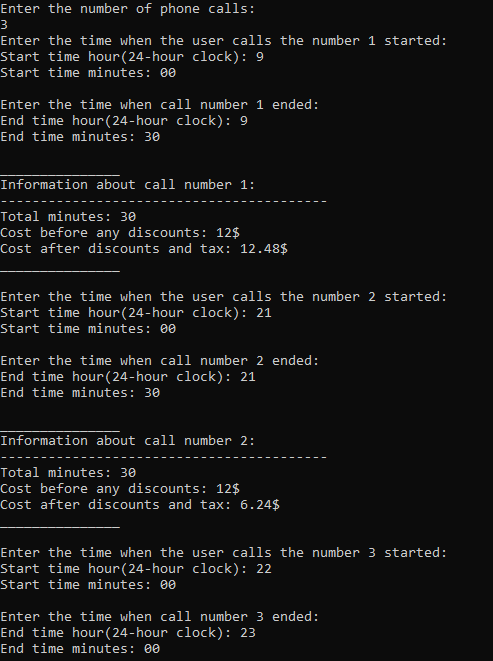
system("pause");

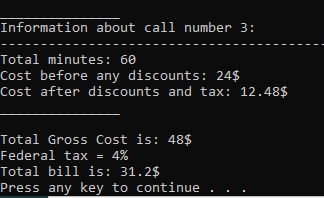
return 0;

}

The aim behind this program is to calculate the telephone bill of a client. This bill is calculated by adding up the net cost of all telephone calls on the bill. The regular rate for a call is $0.40 per minute and all the calls are subjected to 4% federal tax. But if any call is done between 6 P.M (18 on 24-hour clock) and before 8 A.M(8 on 24-hour clock) has a discount 50%. Also, the call that exceeds 60 minutes is subjected to 15% discount. The program should ask the user how many phone calls he/she did, the start time and the end time of the call (based on 24-hour clock). Finally, the program should output for the user the gross cost (before any discounts or tax), net cost (after discounts are deduced and tax is added) for each call and the value of the client bill. But, the value of the client bill is equal to the net cost.

Here we started by applying our program on the compiler after a lot of work on the program. The output will be shown in the following figures:







**Problem 2:**

Write a c++ program to prompt the user to enter his name. The program should ask the user

to guess the login password (generated randomly to be between 1000 – 9999). The user

will try to guess the password digit by digit as follows:

Guess the first digit of the password \*\*\*\*: 2

Sorry wrong guessing!!!.

Guess the first digit of the password \*\*\*\*: 1

WOW you have a correct guess.

Guess the second digit of the password 1\*\*\*: 5

WOW you have a correct guess.

Guess the third digit of the password 15\*\*: 3

Sorry wrong guessing!!!.

.

.

.

The user will have only three chances to guess a specific digit from the password. After 3

successive wrong guesses a $50 balance will be deduced from the user balance. For each

correct guessing a $100 dollar will be added to his balance if it is a correct guess from the

first trial, and a $75 dollar will be added to his balance if it is a correct guess from the second

trial. Note that the user is granted a balance of $200 dollar on the game start.

The user wins a game round if he was able to guess the login password, he loses otherwise.

At the end of the game the user should be prompted about his balance. The program

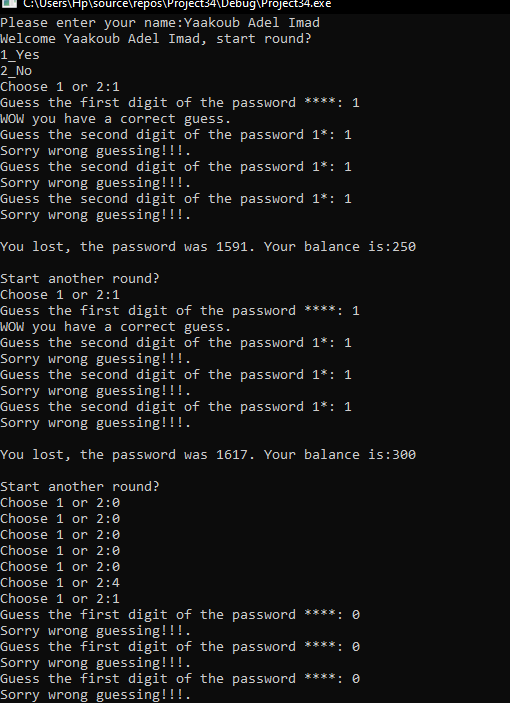
should allow many rounds of the game.

**Program 2:**

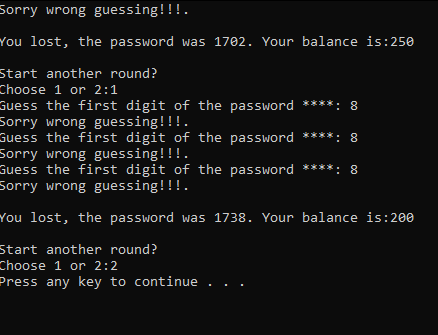
D

The aim behind this program is to make the user to guess the log in pass word which should be generated between 1000 and 9999(made up of 4 digits). This log in password should be entered digit by digit. 50 dollars will be reduced from the balance which is $200 after each three successive wrong guesses, $100 will be added to his balance if it’s a correct guess from the first trial, and $75 will be added to his balance if the user make a correct guess from his second trial. In order for the user to win the round he/she should guess the whole login pass and he/she losses otherwise. Moreover, for the user to have as much rounds as he/she wants, the program asks him if he/she wants to do another round or not and at the end of each round the program prompt his balance.

A function is a code block that only executes when it is invoked. Parameters are data that can be passed into a function. Functions are used to accomplish certain tasks and are essential for code reuse: Once the code is defined, it can be reused numerous times. In this program of problem 2 we only used 1 function called: getFirsyNumber. The getFirstNumber function takes 2 parameters (nb is password and position are the position of the digit from right to left). So, if the user was guessing the first digit, the position will be 4. When the program enters the loop, it will check every time if I is equal to the (position -1) since I started from 0, if not then it will not enter the if condition and will divide the password by 10. It will continue doing this until the i is equal to (position -1), then it will enter the if condition and fill the integer result by password % 10, in order to get the remaining number. When the loop ends, result is returned.



This is the output of the second program. We mostly include all the possible scenarios that would happen.





**Problem 3:**

An integer double-dimensional array called hotel is used to keep track of reservations in a

hotel of 5 floors each floor having 8 rooms. The floors are numbered 1, 2, 3, 4, and 5, and

the rooms are numbered 1, 2, 3… 8. Write a C++ program to manage the hotel reservations

by completing the following:

 Write a function called isFree, that takes the array hotel, the room number and the

floor number, and returns true if that room on that floor is free, and false otherwise.

A reserved room has a value of 1, whereas a free room has a value of 0.

 Write a function called reserveRoom, that takes the array hotel, the room number and

the floor number, and reserves that room on that floor. Note that this function should

use the function isFree to make sure the room to be reserved is free; otherwise the

function should print a message saying the room is already reserved.

 Write a function called freeRoom, that takes the array hotel, the room number and the

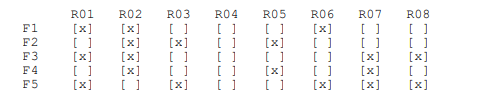
floor number, and frees that room on that floor.

 Write a function called countFree, that takes the array hotel, and returns the number

of free rooms in the hotel.

 Write a function called showHotel, that takes the array hotel, and outputs the status of

all rooms in the hotel. The output should look as follows for the hotel:



Using the functions above, write a program that shows the hotel at the start (all rooms are

free), then allows a user to manage the hotel through a menu (reserve, free, print, etc.).

**The Program of problem 3:**

#include <iostream>

using namespace std;

void fillArray(int arr[][8], int size) // this function is for the user to fill his/her own array.

{

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

{

for (int j = 0; j < 8; j++)

{

arr[i][j] = 0;

}

}

} //end of fillArray function.

bool isFree(int arr[][8], int size, int floor, int room) // to check if the room is free.

{

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

{

for (int j = 0; j < 8; j++)

{

if (i == floor - 1 && j == room - 1)

{

if (arr[i][j])

{

return false;

}

else

{

return true;

}

}

}

}

} //end of isfree function.

void reserveRoom(int arr[][8], int size, int floor, int room) // for the user to reserve the he/she wants.

{

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

{

for (int j = 0; j < 8; j++)

{

if (i == floor - 1 && j == room - 1)

{

if (isFree(arr, size, floor, room))

{

arr[i][j] = 1;

cout << endl;

cout << " ----------------------------------" << endl;

cout << "| The room is reserved successfully|" << endl;

cout << " ----------------------------------" << endl;

cout << endl;

}

else

{

cout << endl;

cout << " ------------------------------" << endl;

cout << "| The room is already reserved |" << endl;

cout << " ------------------------------" << endl;

cout << endl;

}

}

}

}

} // end of reserveroom function.

void freeRoom(int arr[][8], int size, int floor, int room) // to free the room if its already reserved.

{

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

{

for (int j = 0; j < 8; j++)

{

if (i == floor - 1 && j == room - 1)

{

if (isFree(arr, size, floor, room) == false)

{

arr[i][j] = 0;

cout << endl;

cout << " ----------------------" << endl;

cout << "| The room is free now |" << endl;

cout << " ----------------------" << endl;

cout << endl;

}

else

{

cout << endl;

cout << " --------------------------" << endl;

cout << "| The room is already free |" << endl;

cout << " --------------------------" << endl;

cout << endl;

}

}

}

}

}// end of freeroom function.

int countFree(int arr[][8], int size) // to count how many free rooms we have.

{

int free = 0;

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

{

for (int j = 0; j < 8; j++)

{

if (arr[i][j] == 0)

{

free++;

}

}

}

return free;

} // end of countfree function.

void showHotel(int arr[][8], int size) // to show the list of the hotel rooms. All the rooms( free or reserved).

{

cout << "\t R01 \tR02 \tR03 \tR04 \tR05 \tR06 \tR07 \tR08 " << endl;

// cout<<"\t\t R01 \t\t R02 \t\t R03 \t\t R04 \t\t R05 \t\t R06 \t\t R07 \t\t R08 "<<endl;

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

{

cout << "F" << i + 1 << "\t ";

for (int j = 0; j < 8; j++)

{

if (arr[i][j] == 0)

{

cout << "[ ]\t";

}

else

{

cout << "[x]\t";

}

}

cout << endl;

}

cout << endl;

} //end of showhotel function.

int main(int argc, char\*\* argv) { // start of the program.

int floors = 5;

int hotel[5][8];

int choice; // choice of the user.

int roomnb, floornb; // room number and floor number.

int free; // for the free rooms.

fillArray(hotel, floors);

showHotel(hotel, floors);

do { // the menue that the user should choose from.

cout << "1\_Check if room is free." << endl;

cout << "2\_Reserve room." << endl;

cout << "3\_Count free rooms." << endl;

cout << "4\_Check hotel rooms." << endl;

cout << "5\_Free a room." << endl;

cout << "6\_Exit" << endl;

do {

cout << "Choose option from 1 to 6:";

cin >> choice;

} while (choice < 1 || choice > 6);

switch (choice)

{

case 1:

do {

cout << "Enter room floor(1-5):";

cin >> floornb;

} while (floornb < 1 || floornb > 5);

do {

cout << "Enter room number(1-8):";

cin >> roomnb;

} while (roomnb < 1 || roomnb > 8);

if (isFree(hotel, floors, floornb, roomnb))

{

cout << endl;

cout << " --------------" << endl;

cout << "| Room is free |" << endl;

cout << " --------------" << endl;

cout << endl;

}

else

{

cout << endl;

cout << " ------------------" << endl;

cout << "| Room is not free |" << endl;

cout << " ------------------" << endl;

cout << endl;

}

break;

case 2:

do {

cout << "Enter room floor(1-5):";

cin >> floornb;

} while (floornb < 1 || floornb > 5);

do {

cout << "Enter room number(1-8):";

cin >> roomnb;

} while (roomnb < 1 || roomnb > 8);

reserveRoom(hotel, floors, floornb, roomnb);

break;

case 3:

free = countFree(hotel, floors);

cout << endl;

cout << "----------------------------" << endl;

cout << "Number of free rooms: " << free << endl << endl;

cout << "----------------------------" << endl << endl;

break;

case 4:

cout << endl;

showHotel(hotel, floors);

break;

case 5:

do {

cout << "Enter room floor(1-5):";

cin >> floornb;

} while (floornb < 1 || floornb > 5);

do {

cout << "Enter room number(1-8):";

cin >> roomnb;

} while (roomnb < 1 || roomnb > 8);

freeRoom(hotel, floors, floornb, roomnb);

break;

}

} while (choice != 6); // in order for the user to exit the program.

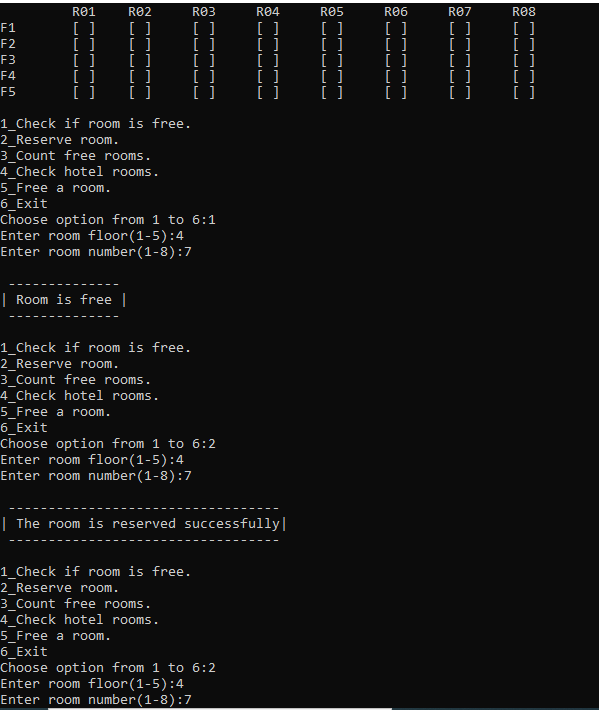
system("pause");

return 0;

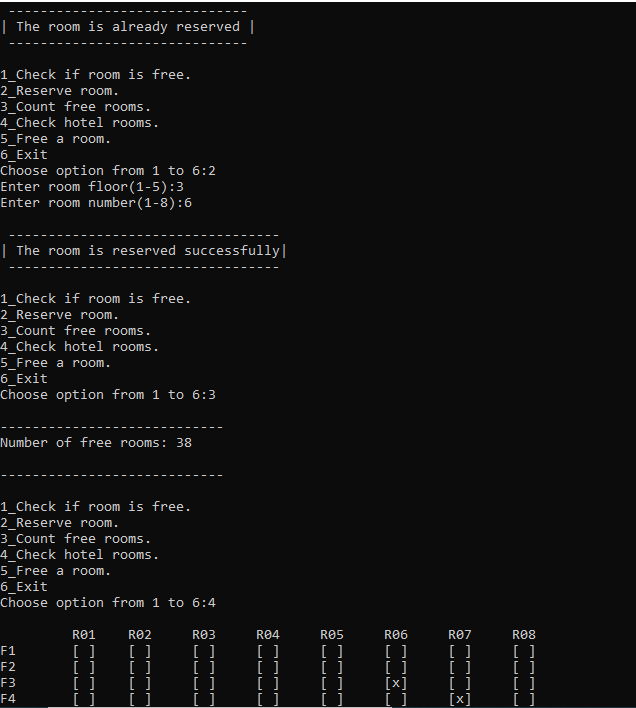
} // end of main.

The aim behind this problem is to write a program that manage the hotel reservations. The hotel is consisted of 5 floors (numbered from 1 to 5) each floor having 8 rooms (numbered from 1 till 8). This program should output the hotel at the start then ask the user to choose from specific menu. This menu involves six options. Neither the user can check if the room is free or he/she can free a room. Also, the user can reserve a room of his/her choice. Moreover, the user can count how many free rooms found in the hotel and check hotel rooms. Finally, the user can exit the program. Note, the user can exit the program at any time he/she needs.

This program contains 6 functions. The first is called fillArray, its use is to let the user to fill his/her hotel design (the number of floors and rooms). The second function is called isFree, it’s a function that takes an array of hotel, room number, and floor number and returns true if that room on that floor is available, and false otherwise. The value of a reserved room is 1, while the value of a free room is 0. The third function is called reserveroom, it’s a function that takes the same array the function isFree takes and reserves that room on that floor. Note that this function should use the isFree function to ensure that the room being reserved is available; otherwise, the function should print a message stating that the room has already been reserved. Freeroom is the fourth function, it’s a function that takes an array of hotel, room number, and floor number and frees that room on that floor. Countfree is the fifth function in the program, it’s a function that takes the hotel array and returns the number of free rooms in the hotel. Finally, the last function is called showhotel, it’s a function that takes an array hotel and returns the status of all rooms in the hotel.

****

Here we tried to apply all the possible choice that would come to the users’ mind.

****

