

**ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION**

**CT018-3-1-ICP**

**INTRODUCTION TO C PROGRAMMING**

**INDIVIDUAL ASSIGNMENT**

**STUDENT NAME: CHAN JIA LE (TP049952)**

**LECTURER: HAIRUL SIVAGURU A/L SUBARMANIYAN**

**INTAKE CODE: UC1F1809CS(DA)**

Contents

[1.0 | Introduction and Assumption 3](#_Toc11241240)

[1.1 | Introduction 3](#_Toc11241241)

[1.2 | Assumption 3](#_Toc11241242)

[2.0 | Program Design 4](#_Toc11241243)

[2.1 | Pseudocode 4](#_Toc11241244)

[2.2 | Flow Chart 22](#_Toc11241245)

[3.0 | C Programming Concept 38](#_Toc11241246)

[3.1 | Header File 38](#_Toc11241247)

[3.2 | Data Structures 38](#_Toc11241248)

[3.3 | Function 39](#_Toc11241249)

[3.4 | Displaying Text 40](#_Toc11241250)

[3.5 | Reading Value 40](#_Toc11241251)

[3.6 | Assigning Value to Variable 40](#_Toc11241252)

[3.7 | Comment 40](#_Toc11241253)

[3.8 | If, Else Statement 41](#_Toc11241254)

[3.9 | Switch, Case 41](#_Toc11241255)

[3.10 | While Loop 41](#_Toc11241256)

[3.11 | For Loop 42](#_Toc11241257)

[3.12 | Open / Close File 42](#_Toc11241258)

[3.13 | Declaration Variable 42](#_Toc11241259)

[3.14 | Read File 43](#_Toc11241260)

[3.15 | Write File 43](#_Toc11241261)

[3.16 | Append File 43](#_Toc11241262)

[3.18 | Strcmp Function 44](#_Toc11241263)

[3.19 | Go to 44](#_Toc11241264)

[3.20 | Array 44](#_Toc11241265)

[3.21 | System (“CLS”) 45](#_Toc11241266)

[3.22 | Sleep Function 45](#_Toc11241267)

[3.23 | Strupr Function 45](#_Toc11241268)

[3.24 | Remove / Rename File 45](#_Toc11241269)

[3.25 | Fscanf Function 45](#_Toc11241270)

[4.0 | Additional Features 46](#_Toc11241271)

[4.1 | Admin Login 46](#_Toc11241272)

[4.2 | Choosing Room 47](#_Toc11241273)

[4.3 | Change Student Personal Details 48](#_Toc11241274)

[4.4 | Show Percentage of Male and Female Hosteler 50](#_Toc11241275)

[5.0 | Test Specification Table 51](#_Toc11241276)

[6.0 | Sample Output: 55](#_Toc11241277)

[6.1 | Hostel Management System Sample Output: 55](#_Toc11241278)

[6.2 | Student Register Sample Output 55](#_Toc11241279)

[6.2.1 | Record Was Found Sample Output 56](#_Toc11241280)

[6.3 | Booking Menu 57](#_Toc11241281)

[6.3.1 | Male Booking Menu 57](#_Toc11241282)

[6.3.2 | Female Booking Menu 57](#_Toc11241283)

[6.4 | Booking Room 58](#_Toc11241284)

[6.4.1 | Male Block A1 Booking 58](#_Toc11241285)

[6.4.2 | Male Block A2 Booking 59](#_Toc11241286)

[6.4.3 | Female Block B1 Booking 60](#_Toc11241287)

[6.4.4 | Female Block B3 Booking 61](#_Toc11241288)

[6.4.5 | Unavailable Block Bookings 62](#_Toc11241289)

[6.5 | Choosing Room Number 63](#_Toc11241290)

[6.5.1 | Room Number Chosen is Occupied 63](#_Toc11241291)

[6.5.2 | Room Number Chosen is Available 63](#_Toc11241292)

[6.5.3 | Choosing Beds in a room 64](#_Toc11241293)

[6.5.4 | Bed Chosen is Occupied 64](#_Toc11241294)

[6.6 | Service Subscription Menu 65](#_Toc11241295)

[6.6.1 | Laundry Subscription Menu 65](#_Toc11241296)

[6.6.2 | Gym Subscription Menu 65](#_Toc11241297)

[6.6.3 | Week of Stay Given & Successful Booking 66](#_Toc11241298)

[6.7 | Student Login 67](#_Toc11241299)

[6.7.1 | No Record of Registration Found 68](#_Toc11241300)

[6.7.2 | Record Found & Successful Login 68](#_Toc11241301)

[6.8 | Student Login Menu 69](#_Toc11241302)

[6.9 | Change/ Cancel Room & Subscribe/ Unsubscribe Laundry or Gym Service 69](#_Toc11241303)

[6.9.1 | Change Room 70](#_Toc11241304)

[6.9.2 | Cancel Room 72](#_Toc11241305)

[6.9.3 | Change Subscription Service 73](#_Toc11241306)

[6.10 | Check / Change Personal Details 74](#_Toc11241307)

[6.11 | Personal Details Menu 75](#_Toc11241308)

[6.11.1 | Change Personal Details 76](#_Toc11241309)

[6.12 | Payment Facility 78](#_Toc11241310)

[6.12.1 | Payment Amount More Than Amount Due 78](#_Toc11241311)

[6.12.2 | Payment Successfully Made 79](#_Toc11241312)

[6.13 | Exit from Student Login System 79](#_Toc11241313)

[6.14 | Admin Login 80](#_Toc11241314)

[6.14.1 | Admin Successfully Login 81](#_Toc11241315)

[6.15 | Admin System 81](#_Toc11241316)

[6.15.1 | Check Overall Details 82](#_Toc11241317)

[6.15.2 | Search for Student Details 83](#_Toc11241318)

[6.15.3 | Check Block Room Availability 84](#_Toc11241319)

[6.15.4 | Exit from Admin System 85](#_Toc11241320)

[6.16 | Exit Program 85](#_Toc11241321)

[7.0 | Conclusion 86](#_Toc11241322)

[8.0 | References 87](#_Toc11241323)

# 1.0 | Introduction and Assumption

## 1.1 | Introduction

Wisdom College is now providing accommodation for its students. There is a total of 8 blocks, 4 blocks for each gender. The Name of the blocks are A1, A2, A3, A4, B1, B2, B3, B4. Block A is for male students and Block B is for female students. Each block contains 100 rooms and the bed quantity is base on the number behind their block. So, Block A1 will be single bed room, block A2 will be double sided bed. Currently only Block A1, A2, B1, B3 is available as the rest are still under construction.

The cost of accommodation is calculated in a weekly basis. The rental for Block A1, B1 will be RM400 per week. Block A2 will be RM200 and Block B3 will be RM150. Students are welcomed to subscribe to additional service as well, the service includes gym service and laundry service, which cost RM10 and RM20 respectively. Students can now use the Hostel Management System to book a hostel room.

## 1.2 | Assumption

The Hostel Management System is built for students to have an easy way to make a booking for hostel as well as designed to allows staff to get easy access over the information. Students are required to register to make a booking. As for those who already has a booking, they should proceed to login into the system.

For new students that is interested to book a room, they will be required to register. The program will then ask the student for their personal details and identify which block they should stay based on the gender they have chosen; the student will then be able to choose single bed room which is block A1 or double side room block A2 as well as choosing their room. The program will display the available rooms allowing for students to choose the room they desire.

After registration process was done, the program will then ask the user to login using their student ID. Students who login will be able to change, cancel room, check or change personal details, make payment and so on. If students want to change or cancel room, they are required to pay a fine of RM200 or RM300 as a compensation for early cancelation.

Admin menu will be able to be accessed by staffs who know the password only, admin menu allows for easy view of overall total payment receivable and received, list all the student details, search for student and check block available rooms.

# 2.0 | Program Design

## 2.1 | Pseudocode

PROGRAM HotelManagementSystem

Create data structure student

string studentID

string name

string gender

integer age

string phoneNumber

string blockArea

integer room number

integer laundry

integer gym

integer weekStay

integer amountDue

integer paidAmount

integer bedNumber

Declare member variable apply and check for student structure

Declare array block\_a1[100]

Declare array block\_a2[100][2]

Declare array block\_b1[100]

Declare array block\_b3[100][3]

BEGIN

counter = 1

DOWHILE (counter = 1)

call initialize\_empty\_room()

call reassign\_previous\_booking()

Print “WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM”

Print “1) STUDENT REGISTER 2) STUDENT LOGIN”

Print “3) ADMIN LOGIN 4)EXIT”

Print “PLEASE ENTER A CHOICE”

Read choice

IF (choice = 1) THEN

gender = call student\_registration()

IF (gender = M) THEN

call male\_booking\_menu()

ELSE

call female\_booking\_menu()

ENDIF

ELSE

IF (choice = 2) THEN

call student\_login()

ENDIF

ELSE

IF (choice = 3) THEN

call admin\_login()

ENDIF

ELSE

IF (choice = 4) THEN

Print “EXIT PROGRAM ”

counter = 0

ENDIF

ELSE

Print “PLEASE ENTER VALID INPUT”

continue

ENDIF

ENDDO

END

FUNCTION initialize\_empty\_room()

LOOP count FROM 0 to 99 STEP 1

block\_a1 [count] = 0

block\_a2 [count][0] = 0

block\_a2 [count][1] = 0

block\_b1 [count] = 0

block\_b3 [count][0] = 0

block\_b3 [count][1] = 0

block\_b3 [count][2] = 0

ENDLOOP

ENDFUNCTION

FUNCTION reassign\_previous\_booking ()

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “A1”) THEN

block\_a1[check.roomNumber] = 1

ELSE

IF (check.blockArea = “A2”) THEN

block\_a2[check.roomNumber][check.bedNumber] = 1

ENDIF

ELSE

IF (check.blockArea = “B1”) THEN

block\_b1[check.roomNumber] = 1’

ENDIF

ELSE

IF (check.blockArea = “B3”) THEN

block\_b3[check.roomNumber][check.bedNumber] = 1

ENDIF

ENDO

CLOSE File “Student.txt”

END FUNCTION

FUNCTION student\_registration()

OPEN File “Student.txt” for Read

Print “STUDENT REGISTRATION SYSTEM”

Print “PLEASE ENTER YOUR STUDENT ID : ”

Read apply.studentID

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Print “YOU HAVE REGISTERED, PLEASE LOGIN !”

RETURN

ENDIF

ENDO

CLOSE File “Student.txt”

Print “PLEASE ENTER YOUR NAME”

Read apply.name

Print “PLEASE ENTER YOUR GENDER (M / F)”

Read apply.gender

Print “PLEASE ENTER YOUR AGE”

Read apply.age

Print “PLEASE ENTER YOUR PHONE NUMBER”

Read apply.phoneNumber

RETURN apply.gender

END FUNCTION

FUNCTION male\_booking\_menu()

OPEN “Student.txt” for append

a1\_available\_room = a1\_room\_available ()

a2\_available\_room = a2\_room\_available ()

Print “BOOKING MENU”

Print “BLOCK A1 AVAILBLE:”

Print a1\_available\_room

Print “BLOCK A2 AVAILABLE”

Print a2\_available\_room

Print “BLOCK A1 = SINGLE ROOM = RM400 WEEKLY”

Print “BLOCK A2 = DOUBLE ROOM = RM200 WEEKLY”

Print “PLEASE INSERT A CHOCE ( 1 – 2 )”

Read block\_type

IF (block\_type = 1) THEN

rental = 400

apply.blockArea = “A1”

room\_num = call a1room\_visual()

laundry = call laundry\_calculation()

gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read apply.weekStay

apply.paidAmount = 0

total\_due = call personal\_due(rental, gym, laundry, apply.weekStay, apply.paidAmount)

apply.room\_num = room\_num

apply.laundry = laundry

apply.gym = gym

apply.amountDue = total\_due

Append data structure student all member variable apply details into “Student.txt”

CLOSE File “Student.txt”

Print “REGISTER SUCCESSFUL”

Print “PLEASE LOGIN !”

ELSE

rental = 200

apply.blockArea = “A2”

room\_num = call a2room\_visual()

laundry = call laundry\_calculation()

gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read apply.weekStay

apply.paidAmount = 0

total\_due = call personal\_due(rental, gym, laundry, apply.weekStay, apply.paidAmount)

apply.room\_num = room\_num

apply.laundry = laundry

apply.gym = gym

apply.amountDue = total\_due

Append data structure student all member variable apply details into “Student.txt”

CLOSE File “Student.txt”

Print “REGISTER SUCCESSFUL”

Print “PLEASE LOGIN !”

ENDIF

END FUNCTION

FUNCTION female\_booking\_menu()

OPEN “Student.txt” for append

b1\_available\_room = b1\_room\_available ()

b3\_available\_room = b3\_room\_available ()

Print “BOOKING MENU”

Print “BLOCK B1 AVAILBLE:”

Print b1\_available\_room

Print “BLOCK B3 AVAILABLE”

Print b3\_available\_room

Print “BLOCK B1 = SINGLE ROOM = RM400 WEEKLY”

Print “BLOCK B3 = TRIPLE ROOM = RM150 WEEKLY”

Print “PLEASE INSERT A CHOCE ( 1 – 3 )”

Read block\_type

IF (block\_type = 1) THEN

rental = 400

apply.blockArea = “B1”

room\_num = call b1room\_visual()

laundry = call laundry\_calculation()

gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read apply.weekStay

apply.paidAmount = 0

total\_due = call personal\_due(rental, gym, laundry, apply.weekStay, apply.paidAmount)

apply.room\_num = room\_num

apply.laundry = laundry

apply.gym = gym

apply.amountDue

Append data structure student all member variable apply details into “Student.txt”

CLOSE File “Student.txt”

Print “REGISTER SUCCESSFUL”

Print “PLEASE LOGIN !”

ELSE

rental = 150

apply.blockArea = “B3”

room\_num = call b3room\_visual()

laundry = call laundry\_calculation()

gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read apply.weekStay

apply.paidAmount = 0

total\_due = call personal\_due(rental, gym, laundry, apply.weekStay, apply.paidAmount)

apply.room\_num = room\_num

apply.laundry = laundry

apply.gym = gym

apply.amountDue

Append data structure student all member variable apply details into “Student.txt”

CLOSE File “Student.txt”

Print “REGISTER SUCCESSFUL”

Print “PLEASE LOGIN !”

ENDIF

END FUNCTION

FUNCTION a1room\_visuals()

z = 0

x = 0

Print (“BLOCK A1”)

LOOP count FROM 0 TO 9 STEP 1

Print “ : ”

LOOP counter FROM 0 to 9 STEP 1

Print “A”, z++

Print block\_a1[x++]

ENDLOOP

Print “”

ENDLOOP

Print “0 MEANS AVAILABLE 1 MEANS OCCUPIED”

Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”

Read room\_num

IF (block\_a1[room\_num] = 0) THEN

block\_a1[room\_num] = 1

ELSE

Print “SORRY THIS ROOM IS OCCUPIED”

ENDIF

RETURN room\_num

END FUNCTION

FUNCTION a2room\_visuals()

z = 0

x = 0

Print (“BLOCK A2”)

LOOP count FROM 0 TO 9 STEP 1

Print “ : ”

LOOP counter FROM 0 to 9 STEP 1

Print “A”, z++

Print block\_a2[x][0], block\_a2[x][1]

x++

ENDLOOP

Print “”

ENDLOOP

Print “0 MEANS AVAILABLE 1 MEANS OCCUPIED”

Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”

Read room\_num

IF (block\_a2[room\_num][0] AND block\_a2[room\_num][1] = 1) THEN

Print “SORRY PLEASE CHOOSE ANOTHER ROOM”

ENDIF

Print “PLEASE CHOOSE A BED (L – R)

Read bed\_choice

IF (bed\_choice = “L” OR bed\_choice = “l” ) THEN

apply.bedNumber = 0

ELSE

IF (bed\_choice = “R” OR bed\_choice = “r” ) THEN

apply.bedNumber = 1

ENDIF

ENDIF

IF (block\_a2[room\_num][apply.bedNumber] =1 ) THEN

Print “SORRY THIS BED IS OCCUPIED”

ELSE

IF (block\_a2[room\_num][0] OR block\_a2[room\_num][1] = 0)

block\_a2[room\_num][apply.bedNumber] = 1

ENDIF

ENDIF

RETURN room\_num

END FUNCTION

FUNCTION b1room\_visuals()

z = 0

x = 0

Print (“BLOCK B1”)

LOOP count FROM 0 TO 9 STEP 1

Print “ : ”

LOOP counter FROM 0 to 9 STEP 1

Print “B”, z++

Print block\_b1[x++]

ENDLOOP

Print “”

ENDLOOP

Print “0 MEANS AVAILABLE 1 MEANS OCCUPIED”

Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”

Read room\_num

IF (block\_b1[room\_num] = 0) THEN

block\_b1[room\_num] = 1

ELSE

Print “SORRY THIS ROOM IS OCCUPIED”

ENDIF

RETURN room\_num

END FUNCTION

FUNCTION b3room\_visuals()

z = 0

x = 0

Print (“BLOCK B3”)

LOOP count FROM 0 TO 9 STEP 1

Print “ : ”

LOOP count FROM 0 to 9 STEP 1

Print “B”, z++

Print block\_b3[x][0], block\_b3[x][1],block\_b3[x][2]

x++

ENDLOOP

Print “”

ENDLOOP

Print “0 MEANS AVAILABLE 1 MEANS OCCUPIED”

Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”

Read room\_num

IF (block\_b3[room\_num][0] AND block\_b3[room\_num][1] = 1 AND block\_b3[room\_num][2] = 1) THEN

Print “SORRY PLEASE CHOOSE ANOTHER ROOM”

ENDIF

Print “PLEASE CHOOSE A BED (L – M – R)

Read bed\_choice

IF (bed\_choice = “L” OR bed\_choice = “l” ) THEN

apply.bedNumber = 0

ELSE

IF (bed\_choice = “M” OR bed\_choice = “m” ) THEN

apply.bedNumber = 1

ENDIF

ELSE

IF (bed\_choice = “R” OR bed\_choice = “r” ) THEN

apply.bedNumber = 2

ENDIF

ENDIF

IF (block\_a2[room\_num][apply.bedNumber] =1 ) THEN

Print “SORRY THIS BED IS OCCUPIED”

ELSE

IF (block\_a2[room\_num][0] OR block\_a2[room\_num][1] = 0) THEN

block\_a2[room\_num][apply.bedNumber] = 1

ENDIF

ENDIF

RETURN room\_num

END FUNCTION

FUNCTION a1\_room\_available()

i = 0

occupied\_room = 0

DOWHILE ( i < 100)

IF (block\_a1[i] = 1)

occupied\_room++

ENDIF

i ++

ENDDO

available\_room = 100 – occupied\_room

RETURN available\_room

END FUNCTION

FUNCTION a2\_room\_available()

i = 0

occupied\_room = 0

DOWHILE ( i < 100)

IF ((block\_a2[i][0] AND block\_a2[i][1]) = 1) THEN

occupied\_room++

ENDIF

i ++

ENDDO

available\_room = 100 – occupied\_room

RETURN available\_room

END FUNCTION

FUNCTION b1\_room\_available()

i = 0

occupied\_room = 0

DOWHILE ( i < 100)

IF (block\_b1[i] = 1) THEN

occupied\_room++

ENDIF

i ++

ENDDO

available\_room = 100 – occupied\_room

RETURN available\_room

END FUNCTION

FUNCTION b3\_room\_available()

i = 0

occupied\_room = 0

DOWHILE ( i < 100)

IF ((block\_b3[i][0] AND block\_b3[i][1] AND block\_b3[i][2]) = 1) THEN

occupied\_room++

ENDIF

i ++

ENDDO

available\_room = 100 – occupied\_room

RETURN available\_room

END FUNCTION

FUNCTION laundry\_calculation()

Print “LAUNDRY = RM20”

Print “DO YOU WANT TO ADD LAUNDRY SERVICE”

Print “1. YES 2.NO”

Print “PLEASE ENTER A CHOICE (1 - 2)”

Read laundry\_choice

IF (laundry\_choice = 1) THEN

laundry = 20

RETURN laundry

ELSE

laundry = 0

RETURN laundry

ENDIF

END FUNCTION

FUNCTION gym\_calculation()

Print “ GYM RM10”

Print “DO YOU WANT TO ADD GYM SERVICE”

Print “1. YES 2.NO”

Print “PLEASE ENTER A CHOICE (1 - 2)”

Read gym\_choice

IF (gym\_choice = 1) THEN

gym = 10

RETURN gym

ELSE

gym = 0

RETURN gym

ENDIF

END FUNCTION

FUNCTION personal\_due( rental, gym, laundry, week\_stay, paid\_amount)

total\_personal\_amount = rental + gym + laundry

total\_personal\_amount = (total\_personal\_amount \* week\_stay) – paid\_amount

RETURN total\_personal\_amount

END FUNCTION

FUNCTION student\_login()

Print “STUDENT LOGIN SYSTEM”

Print “PLEASE ENTER YOUR STUDENT ID”

Read apply.studentID

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Print “SUCCESSFULLY LOGIN WITH STUDENT ID”

check\_login = 1

BREAK

ELSE

check\_login = 0

ENDIF

ENDDO

IF (check\_login = 1) THEN

Print “WELCOME TO STUDENT LOGIN SYSTEM”

Print “WHAT WOULD YOU LIKE TO DO”

Print “1. CHANGE OR CANCEL ROOM 2. CHECK PERSONAL DETAIL”

Print “3. MAKE PAYMENT 4. EXIT TO MAIN MENU”

Read choice

IF (choice = 1) THEN

Print “WOULD YOU LIKE TO 1. CHANGE 2.CANCEL ROOM”

Print “PLEASE ENTER A CHOICE”

Read change\_choice

IF (change\_choice = 1) THEN

call change\_room()

ELSE

IF (change\_choice = 2) THEN

call cancel\_room()

ENDIF

ENDIF

ELSE

IF (choice = 2) THEN

call student\_details()

ENDIF

ELSE

IF (choice = 3) THEN

call payment\_facility()

ENDIF

ELSE

IF (choice = 4) THEN

RETURN

ENDIF

ENDIF

ELSE

Print “RETURNING TO MAIN MENU”

RETURN

ENDIF

CLOSE File “Student.txt”

END FUNCTION

FUNCTION change\_room()

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF ((apply.studentID = check.studentID) AND check.gender = “M”) THEN

CLOSE File “Student.txt”

a1\_available\_room = a1\_room\_available ()

a2\_available\_room = a2\_room\_available ()

Print “BOOKING MENU”

Print “BLOCK A1 AVAILBLE:”

Print a1\_available\_room

Print “BLOCK A2 AVAILABLE”

Print a2\_available\_room

Print “BLOCK A1 = SINGLE ROOM = RM400 WEEKLY”

Print “BLOCK A2 = DOUBLE ROOM = RM200 WEEKLY”

Print “PLEASE INSERT A CHOCE ( 1 – 2 )”

Read block\_type

IF (block\_type = 1) THEN

new\_rental = 400

check.blockArea = “A1”

new\_room\_num = call a1room\_visual()

new\_laundry = call laundry\_calculation()

new\_gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read new\_weekStay

new\_paidAmount = 0

new\_total\_due = call personal\_due(new\_rental, new\_gym, new\_laundry, new\_weekStay, new\_paidAmount)

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File “Student.txt”and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID)

Write new data inside text file “NewStudent.txt”

ELSE

Write previous data inside text file “NewStudent.txt”

ENDIF

ENDDO

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

Print “ CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN”

ELSE

new\_rental = 200

check.blockArea = “A2”

new\_room\_num = call a2room\_visual()

new\_laundry = call laundry\_calculation()

new\_gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read new\_weekStay

new\_paidAmount = 0

new\_total\_due = call personal\_due(new\_rental, new\_gym, new\_laundry, new\_weekStay, new\_paidAmount)

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File “Student.txt”and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Write new data inside text file “NewStudent.txt”

ELSE

Write previous data inside text file “NewStudent.txt”

ENDIF

ENDDO

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

Print “ CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN”

ENDIF

ELSE

IF ((apply.studentID = check.studentID) AND check.gender = “F”) THEN

CLOSE File “Student.txt”

b1\_available\_room = b1\_room\_available ()

b3\_available\_room = b3\_room\_available ()

Print “BOOKING MENU”

Print “BLOCK B1 AVAILBLE:”

Print a1\_available\_room

Print “BLOCK B3 AVAILABLE”

Print b3\_available\_room

Print “BLOCK B1 = SINGLE ROOM = RM400 WEEKLY”

Print “BLOCK B3 = TRIPLE ROOM = RM150 WEEKLY”

Print “PLEASE INSERT A CHOCE ( 1 – 2 )”

Read block\_type

IF (block\_type = 1) THEN

new\_rental = 400

check.blockArea = “A1”

new\_room\_num = call a1room\_visual()

new\_laundry = call laundry\_calculation()

new\_gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read new\_weekStay

new\_paidAmount = 0

new\_total\_due = call personal\_due(new\_rental, new\_gym, new\_laundry, new\_weekStay, new\_paidAmount)

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File “Student.txt”and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Write new data inside text file “NewStudent.txt”

ELSE

Write previous data inside text file “NewStudent.txt”

ENDIF

ENDDO

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

Print “ CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN”

ELSE

new\_rental = 150

check.blockArea = “B3”

new\_room\_num = call b3room\_visual()

new\_laundry = call laundry\_calculation()

new\_gym = call gym\_calculation()

Print “HOW MANY WEEKS WOULD YOU LIKE TO STAY?”

Read new\_weekStay

new\_paidAmount = 0

new\_total\_due = call personal\_due(new\_rental, new\_gym, new\_laundry, new\_weekStay, new\_paidAmount)

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File “Student.txt”and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Write new data inside text file “NewStudent.txt”

ELSE

Write previous data inside text file “NewStudent.txt”

ENDIF

ENDDO

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

Print “ CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN”

ENDIF

ENDIF

ENDIF

ENDDO

END FUNCTION

FUNCTION cancel\_room()

Print “ARE YOU SURE YOU WANT TO CANCEL ROOM? Y OR N”

Read cancel\_choice

IF (cancel\_choice = “Y”) THEN

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

CONTINUE

ELSE

Write previous data inside text file “NewStudent.txt”

ENDIF

ENDDO

Print “CANCEL ROOM SUCCESSFUL”

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

ELSE

IF (cancel\_choice = “N”) THEN

RETURN

ENDIF

ENDIF

END FUNCTION

FUNTION student\_details()

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

Print “STUDENT ID: ”, check.studentID

Print “NAME:”, check.name

Print “GENDER: ” , check.gender

Print “AGE: ”, check.age

Print “PHONE :” , check.phoneNumber

Print “BLOCK :”, check.blockArea

Print “ROOM NUMBER”, check.room\_number

Print “WEEK OF STAY: ”, check.weekStay

Print “AMOUNT DUE:”, check.amountDue

Print “PAID AMOUNT:”, check.paidAmount

Print “HERE ARE YOUR DETAILS, ENTER ANYTHING TO EXIT :”

Read choice

ENDIF

ENDDO

CLOSE File “Student.txt”

END FUNCTION

FUNCTION payment\_facility()

Print “PAYMENT FACILITY”

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

payment\_due = check.amountDue

ENDIF

ENDDO

CLOSE File “Student.txt”

Print “HI , YOUR PAYMENT DURE IS ”, payment\_due

Print “HOW MUCH WOULD YOU LIKE TO PAY?”

Read payment\_made

OPEN File “Student.txt” for Read

OPEN File “NewStudent.txt” for Write

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

new\_payment\_due = payment\_due – payment\_made

new\_payment\_made = check.paidAmount + payment\_made

Write new\_payment\_due and new\_payment\_made the rest of student data to text file “NewStudent.txt”

ELSE

Write previous data into text file “NewStudent.txt”

ENDIF

ENDDO

Print “PAYMENT IS SUCCESSFUL”

CLOSE File “Student.txt”

CLOSE File “NewStudent.txt”

REMOVE File “Student.txt”

RENAME File “NewStudent.txt” to “Student.txt”

END FUNCTION

FUNCTION admin\_login()

password = 1234

Print “ADMIN LOGIN SYSTEM”

Print “PLEASE ENTER PASSWORD”

Read check\_password

IF (check\_password = 1234) THEN

call admin\_system()

ELSE

Print “WRONG PASSWORD PLEASE TRY AGAIN”

RETURN

ENDIF

END FUNCTION

FUNCTION admin\_system()

Print “WELCOME TO ADMIN SYSTEM”

Print “1) CHECK OVERALL DETAILS 2) SEACH FOR STUDENT DETAILS”

Read admin\_choice

IF (admin\_choice = 1) THEN

call check\_r\_available\_and\_m\_receivables()

ELSE

IF (admin\_choice = 2) THEN

call search\_student()

ENDIF

ELSE

RETURN

ENDIF

END FUNCTION

FUNCTION check\_r\_available\_and\_m\_receivables()

OPEN File “Student.txt” for Read

a1\_room\_count = call a1\_room\_avalable()

a2\_room\_count = call a2\_room\_avalable(),

b1\_room\_count = call b1\_room\_avalable(),

b3\_room\_count = call b3\_room\_avalable(),

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “A1”) THEN

a1\_amount\_receivable = a1\_amount\_receivable + check.amountDue

a1\_payment\_received = a1\_payment\_received + check.paidAmount

a1\_student\_amount++

ELSE

IF (check.blockArea = “A2”) THEN

a2\_amount\_receivable = a2\_amount\_receivable + check.amountDue

a2\_payment\_received = a2\_payment\_received + check.paidAmount

a2\_student\_amount++

ENDIF

ELSE

IF (check.blockArea = “B1”) THEN

b1\_amount\_receivable = b1\_amount\_receivable + check.amountDue

b1\_payment\_received = b1\_payment\_received + check.paidAmount

b1\_student\_amount++

ENDIF

ELSE

b3\_amount\_receivable = b3\_amount\_receivable + check.amountDue

b3\_payment\_received = b3\_payment\_received + check.paidAmount

b3\_student\_amount++

ENDIF

ENDDO

total\_amount\_recivable = a1\_amount\_receivable + a2\_amount\_receivable+ b1\_amount\_receivable+ b3\_amount\_receivable

total\_room\_available = a1\_room\_count + a2\_room\_count + b1\_room\_count + b3\_room\_count

total\_amount\_payment = a1\_payment\_received + a2\_payment\_received + b1\_payment\_received + b3\_payment\_received

total\_student\_amount = a1\_student\_amount + a2\_student\_amount + b1\_student\_amount + b3\_student\_amount

Print “Block A1 Available Room :”, a1\_room\_count

Print “Block A2 Available Room :”, a2\_room\_count

Print “Block B1 Available Room :”, b1\_room\_count

Print “Block B3 Available Room :”, b3\_room\_count

Print “Total Room Available :”, total\_room\_available

Print “Total Amount Receivable :”, total\_amount\_receivable

Print “Total Payment Received:”, total\_amount\_payment

Print “If you want to check all the student details including amount receivable input 1”

Read check\_choice

IF (check\_choice = 1) THEN

call check\_block\_student()

ELSE

Print “INVALID INPUT”

RETURN

ENDIF

CLOSE File “Student.txt”

ENDFUNCTION

FUNCTION check\_block\_student()

OPEN File “Student.txt” for Read

Print “PLEASE CHOOSE THE BLOCK YOU WANT TO VIEW (A1, A2, B1,B3)”

Read check\_block

IF (check\_block = “A1”) THEN

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “A1”)

Print “STUDENT ID:”,check.studentID

Print “NAME:”,check.name

Print “AMOUNT RECEIVABLE: ”, check.amountDue

ENDIF

ENDDO

ELSE

IF (check\_block = “A2”) THEN

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “A2”)

Print “STUDENT ID:”,check.studentID

Print “NAME:”,check.name

Print “AMOUNT RECEIVABLE: ”, check.amountDue

ENDIF

ENDDO

ENDIF

ELSE

IF (check\_block = “B1”) THEN

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “B1”)

Print “STUDENT ID:”,check.studentID

Print “NAME:”,check.name

Print “AMOUNT RECEIVABLE: ”, check.amountDue

ENDIF

ENDDO

ENDIF

ELSE

IF (check\_block = “B3”) THEN

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (check.blockArea = “B3”)

Print “STUDENT ID:”,check.studentID

Print “NAME:”,check.name

Print “AMOUNT RECEIVABLE: ”, check.amountDue

ENDIF

ENDDO

ENDIF

ENDIF

CLOSE File “Student.txt”

Print “PRESS ANYTHING TO EXIT”

Read exit\_choice

RETURN

END FUNCTION

FUNCTION search\_students()

Print “STUDENT SEARCH SYSTEM”

Print “PLEASE INSERT THE STUDENT ID TO SEARCH FOR STUDENT DETAILS :”

Read search\_studentID

OPEN File “Student.txt” for Read

DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)

IF (apply.studentID = check.studentID) THEN

check\_login = 1

BREAK

ELSE

check\_login = 0

ENDIF

ENDDO

IF (check\_login = 0) THEN

Print (“NO RECORD WAS FOUND”)

RETURN

ELSE

Print “BOOKING DETAILS OF SEARCH STUDENT”

Print “STUDENT ID: ”, check.studentID

Print “NAME:”, check.name

Print “GENDER: ” , check.gender

Print “AGE: ”, check.age

Print “PHONE :” , check.phoneNumber

Print “BLOCK :”, check.blockArea

Print “ROOM NUMBER”, check.room\_number

Print “WEEK OF STAY: ”, check.weekStay

Print “AMOUNT DUE:”, check.amountDue

Print “PAID AMOUNT:”, check.paidAmount

Print “HERE ARE YOUR DETAILS, ENTER ANYTHING TO EXIT :”

Read choice

RETURN

ENDIF

END FUNCTION

## 2.2 | Flow Chart

Figure 2.2.2 Initialize Empty Room Function

Figure 2.2.1 Hostel Management Main Menu

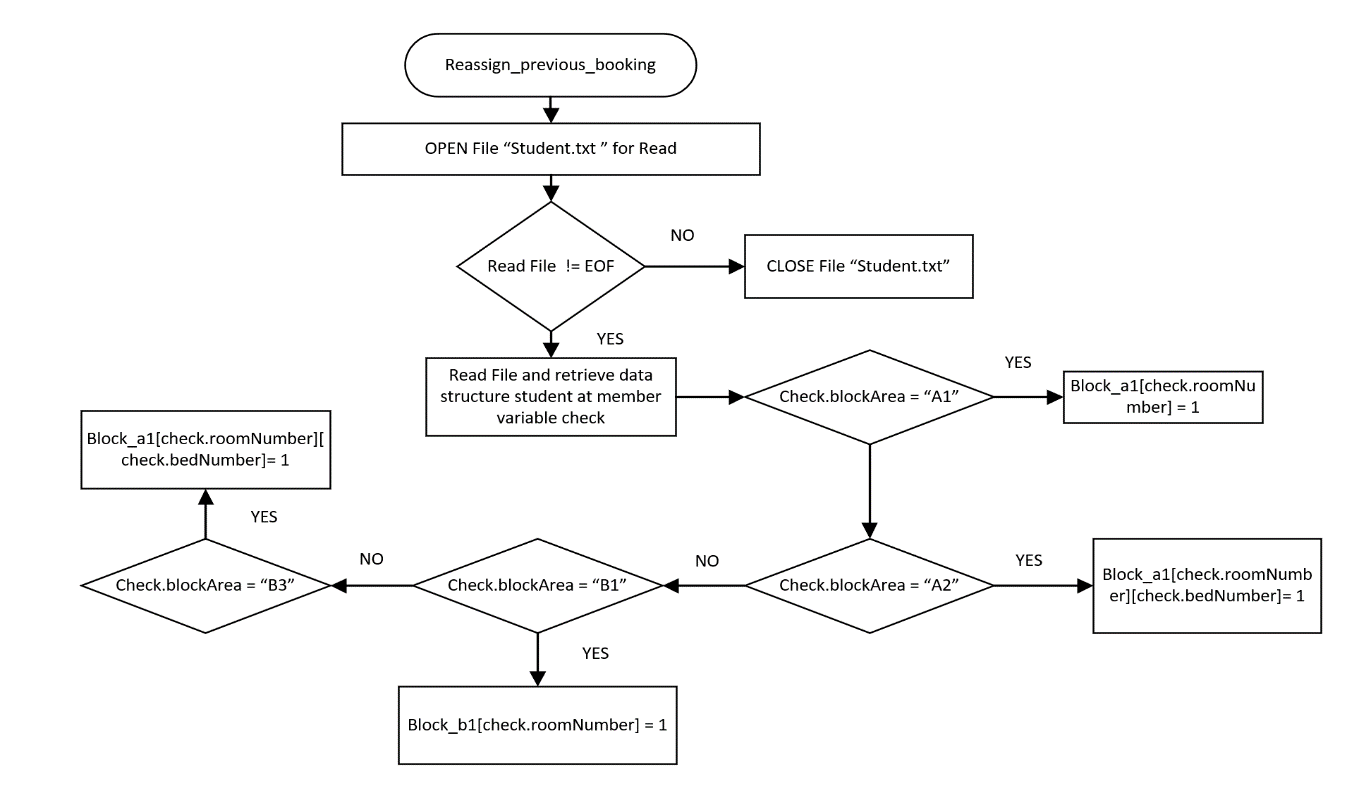
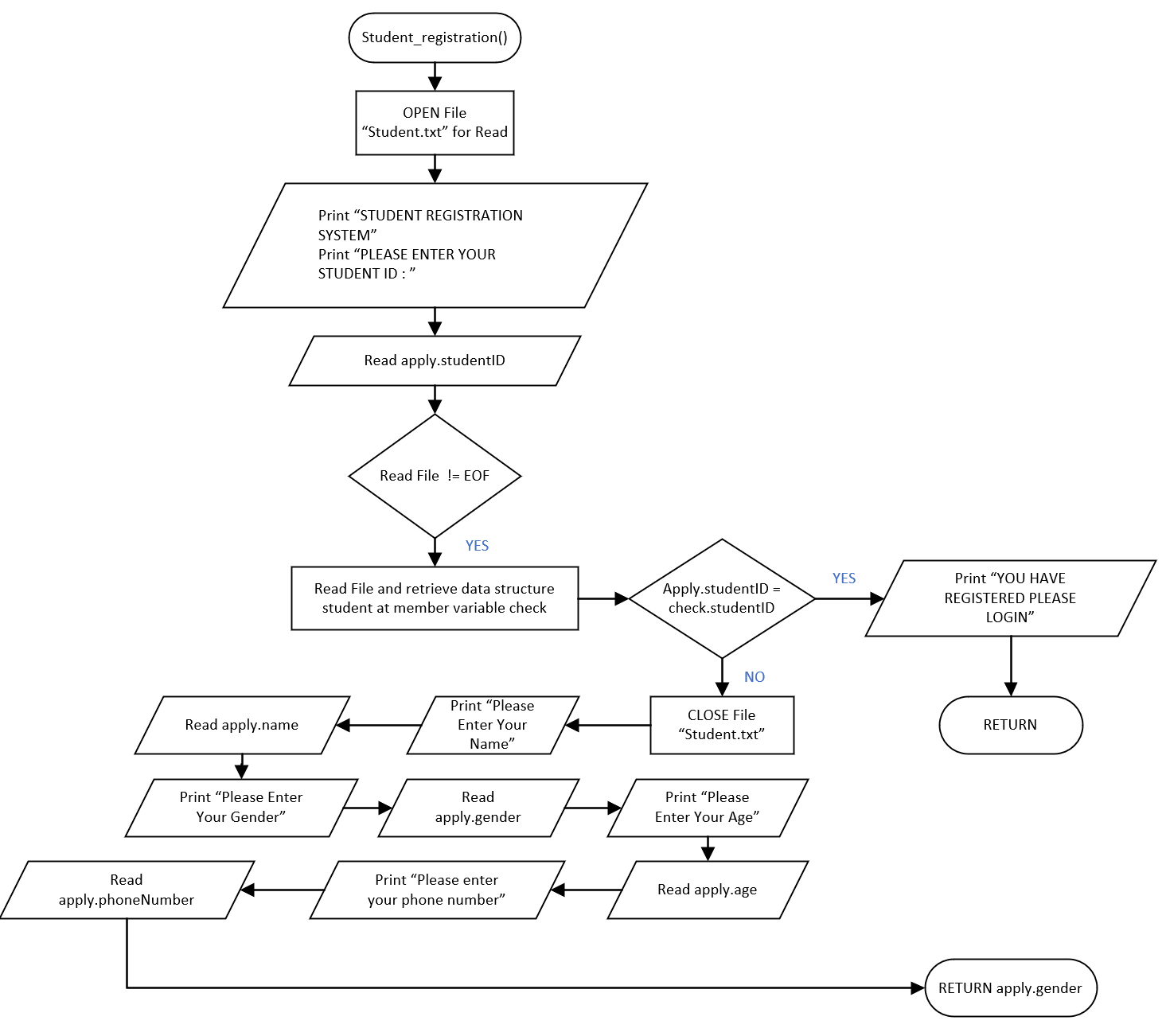


Figure 2.2.4 Student Registration Function

Figure 2.2.3 Reassign Previous Booking Function

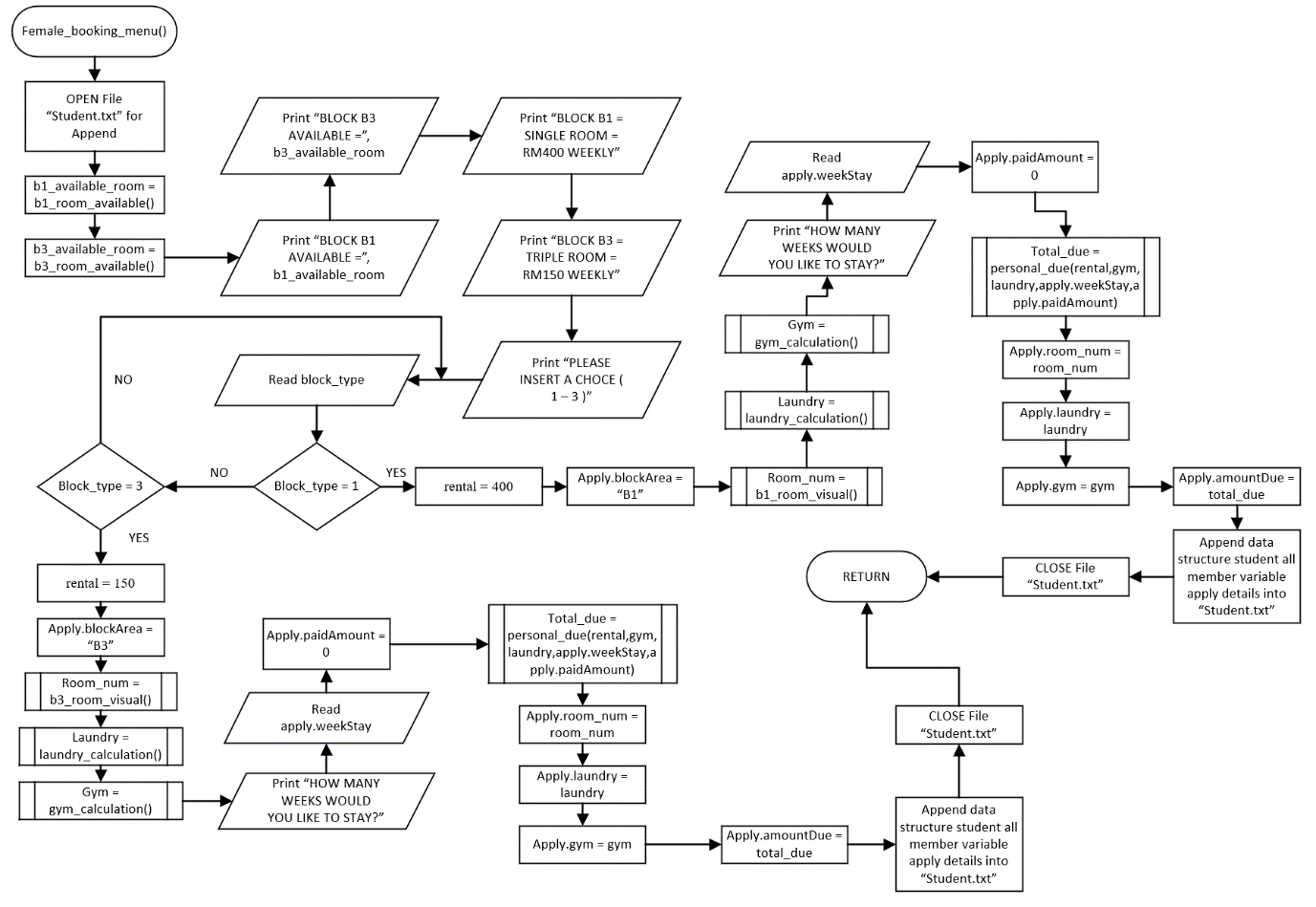
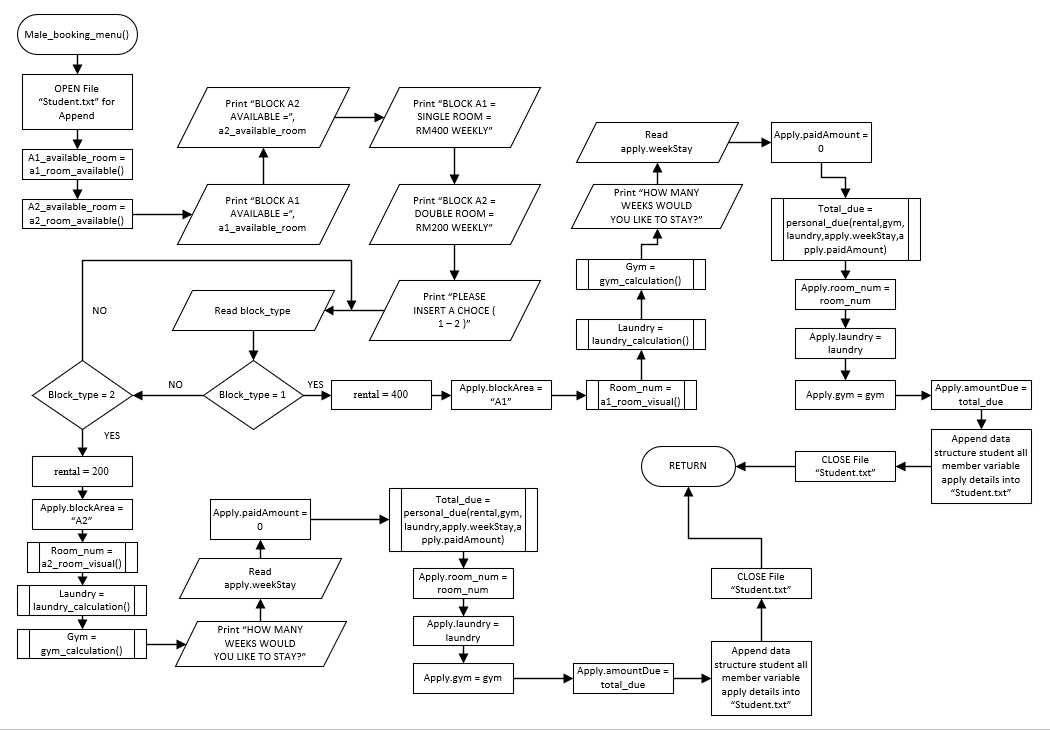
 

Figure 2.2.6 Female Booking Menu Function

Figure 2.2.5 Male Booking Menu Function

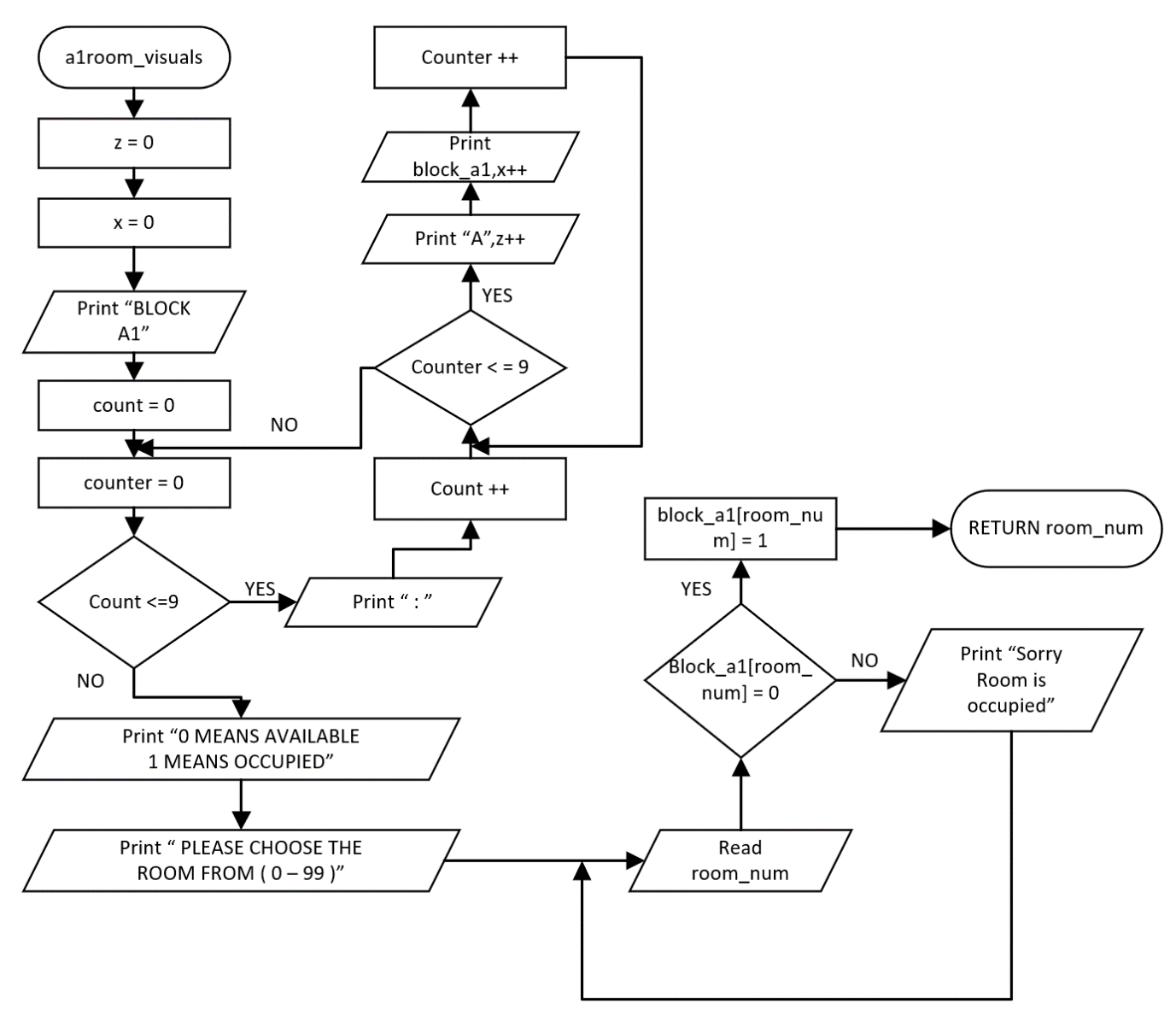
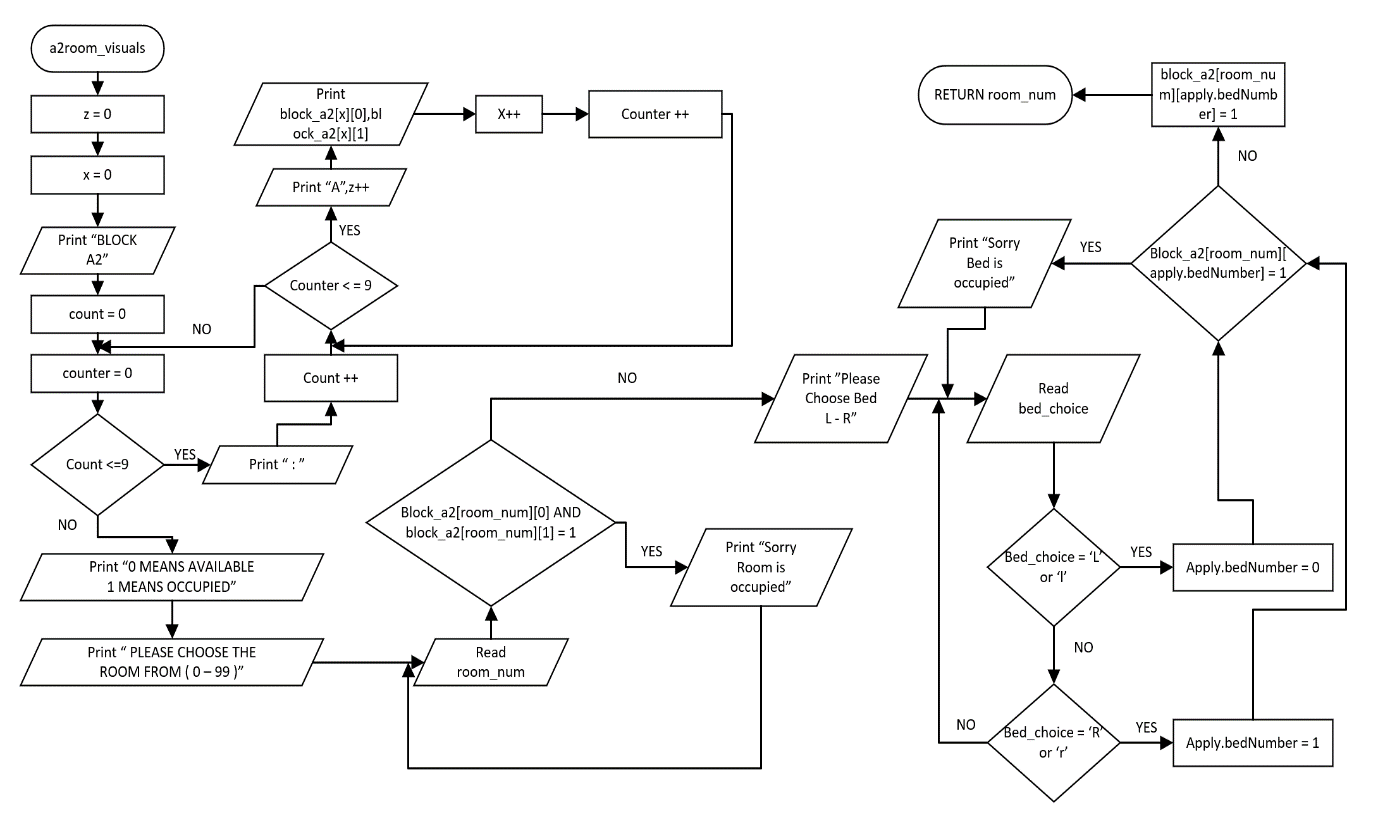


Figure 2.2.8 A2 Room Visual Function

Figure 2.2.7 A1 Room Visual Function

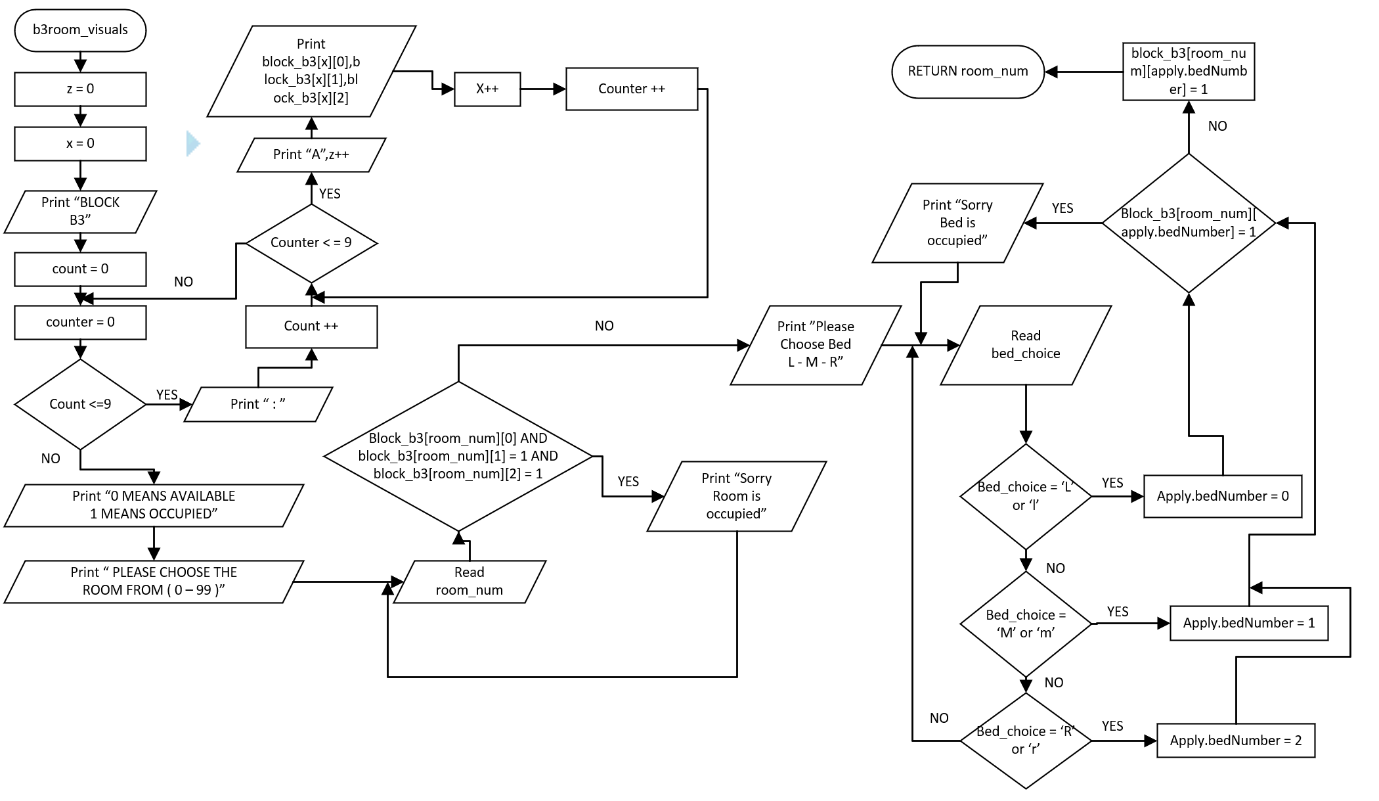
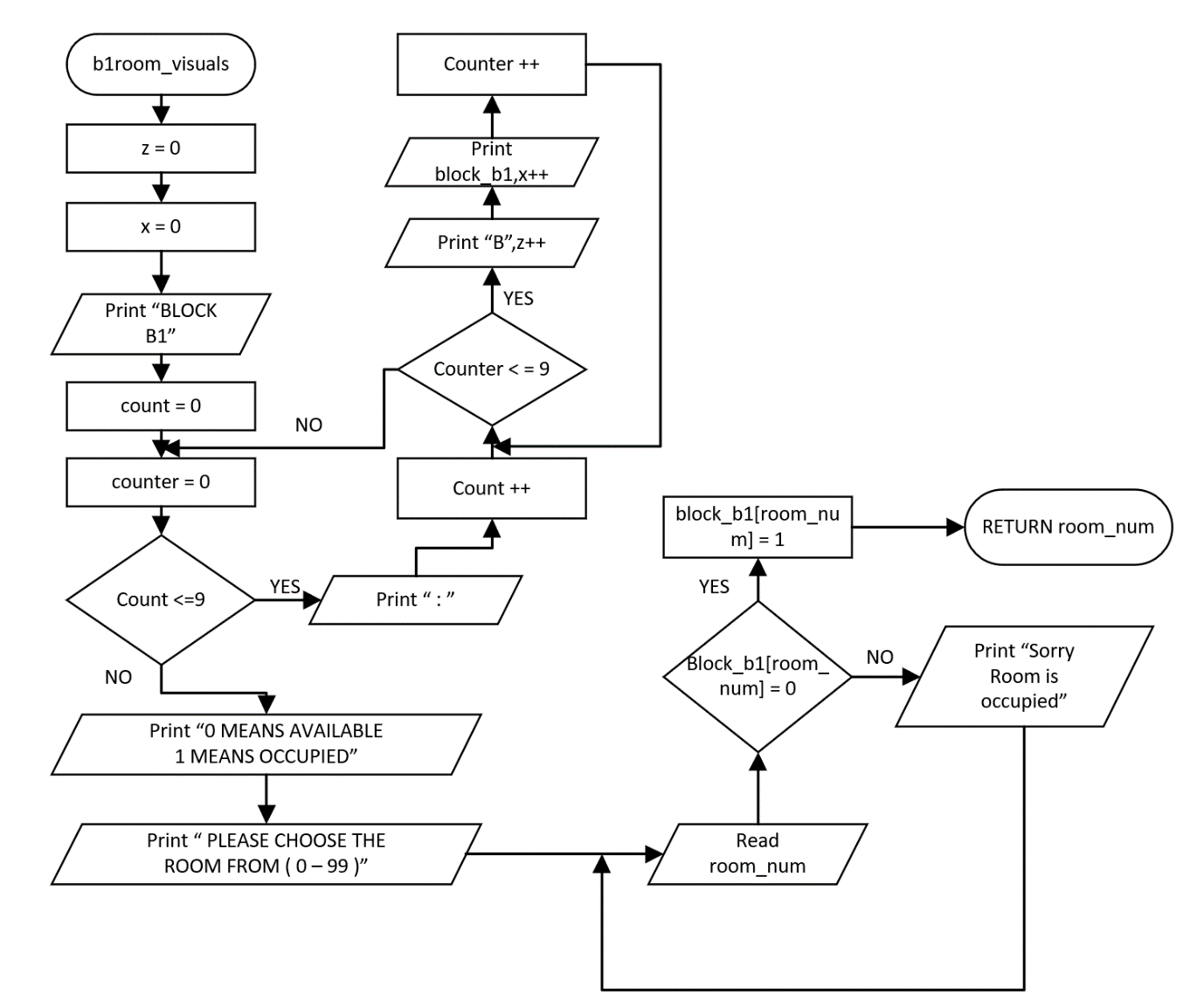
 

Figure 2.2.10 B3 Room Visuals

Figure 2.2.9 B1 Room Visuals

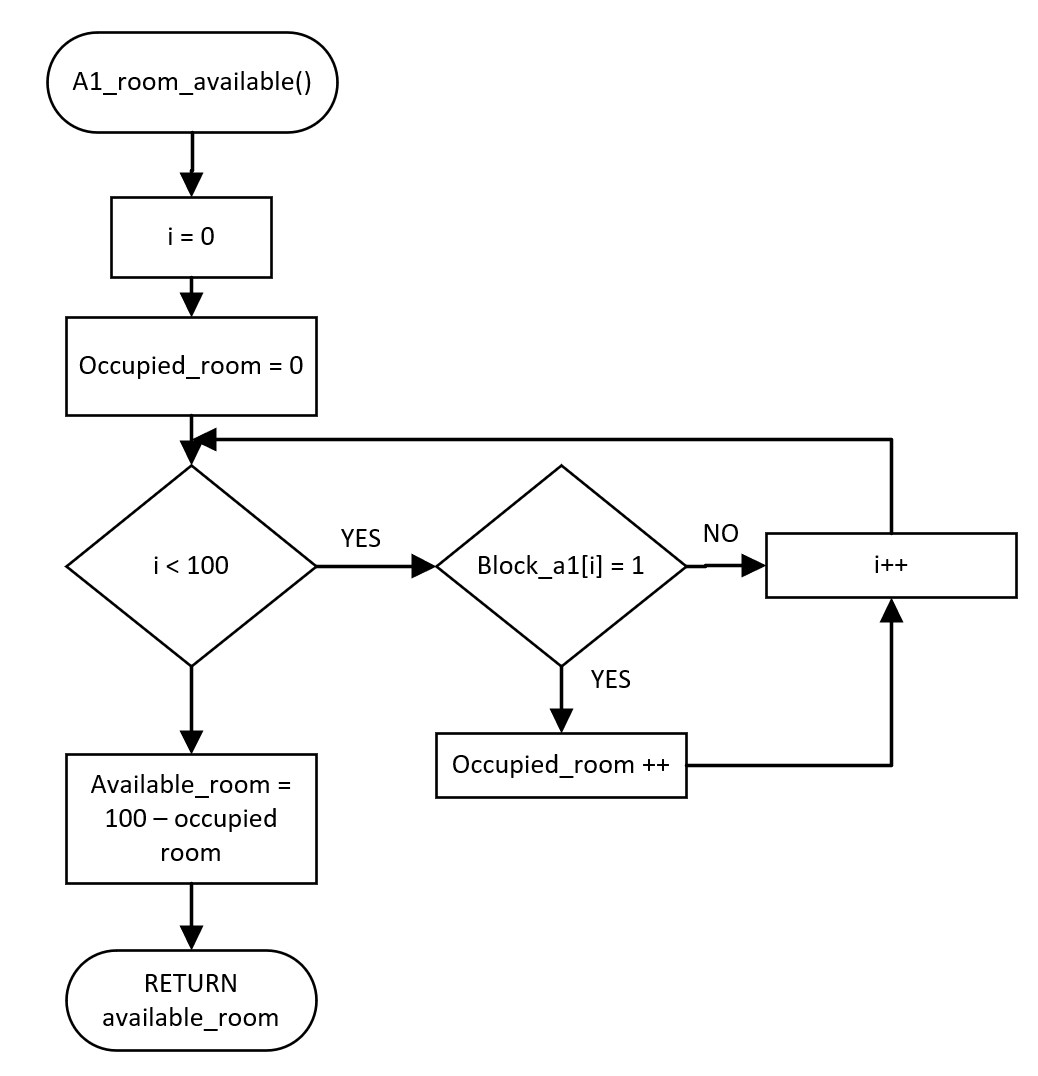


Figure 2.2.11 A1 Room Available Function

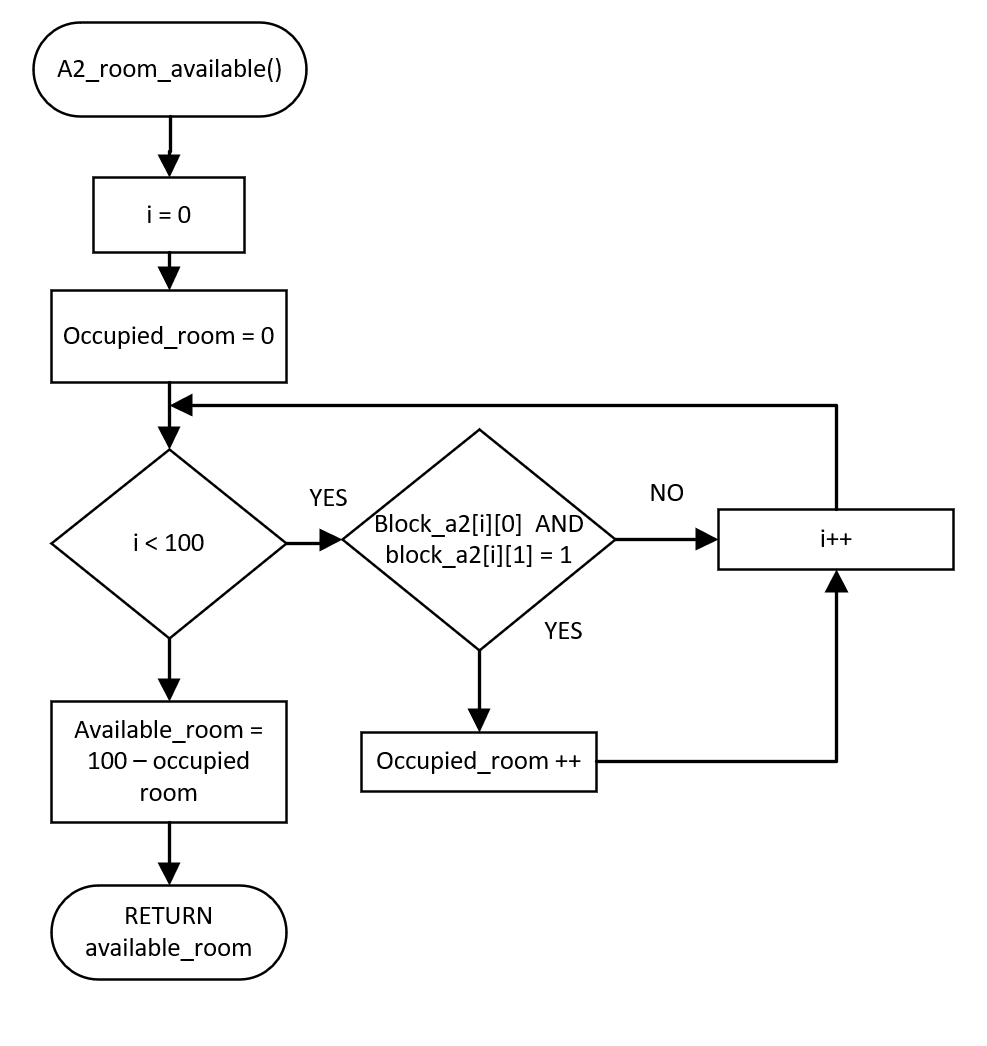


Figure 2.2.12 A2 Room Available Function

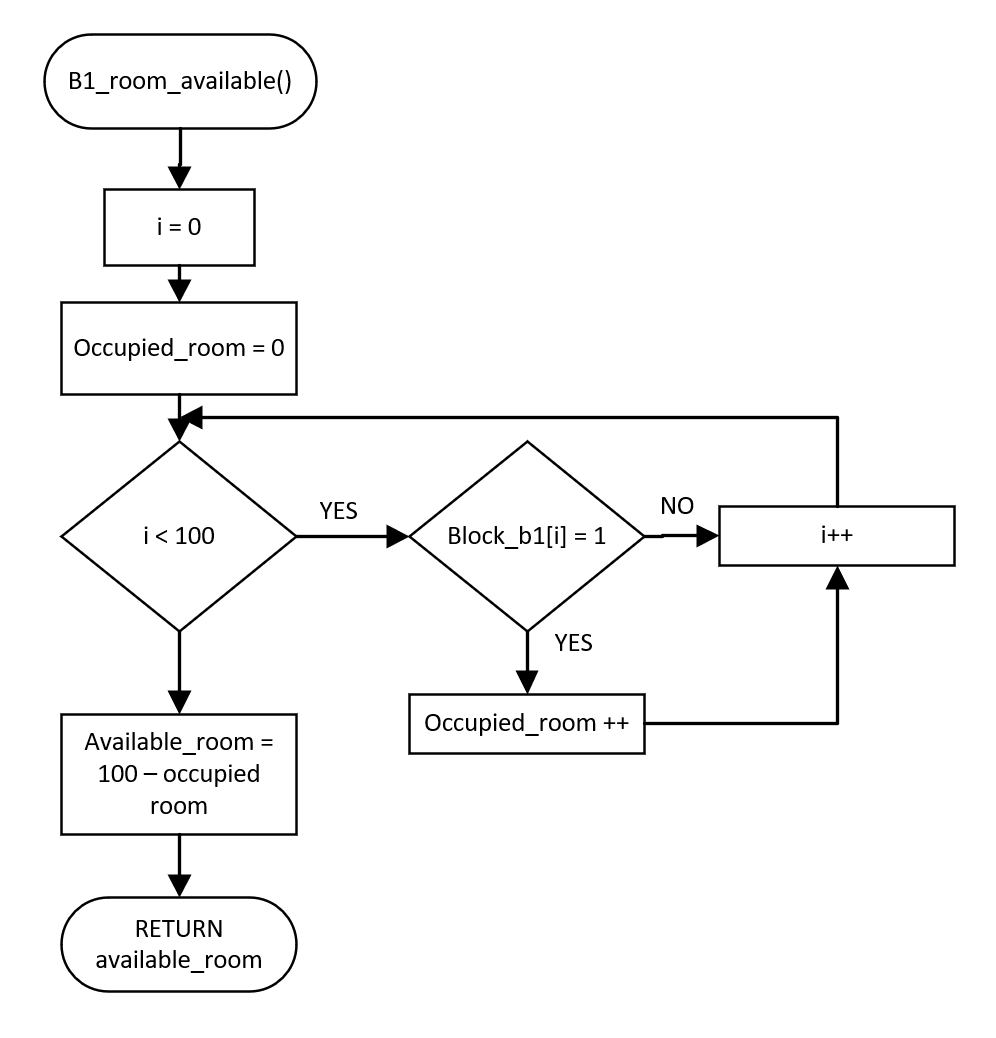


Figure 2.2.13 B1 Room Available Function

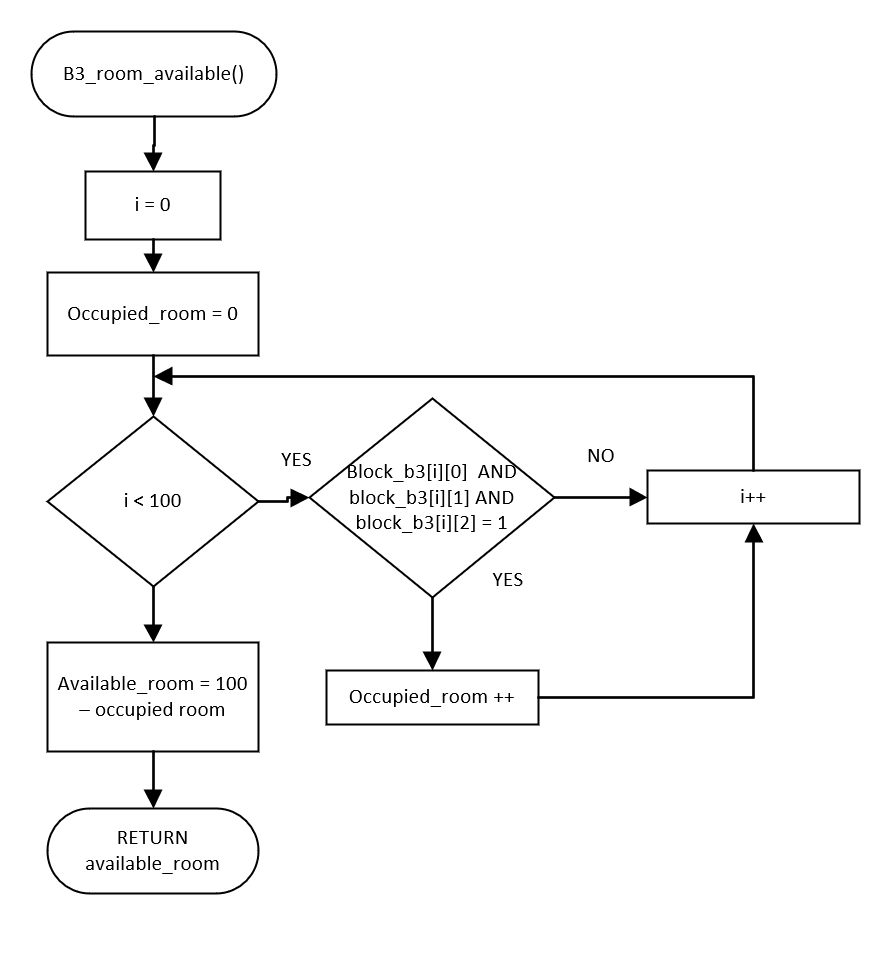


Figure 2.2.14 B3 Room Available Function

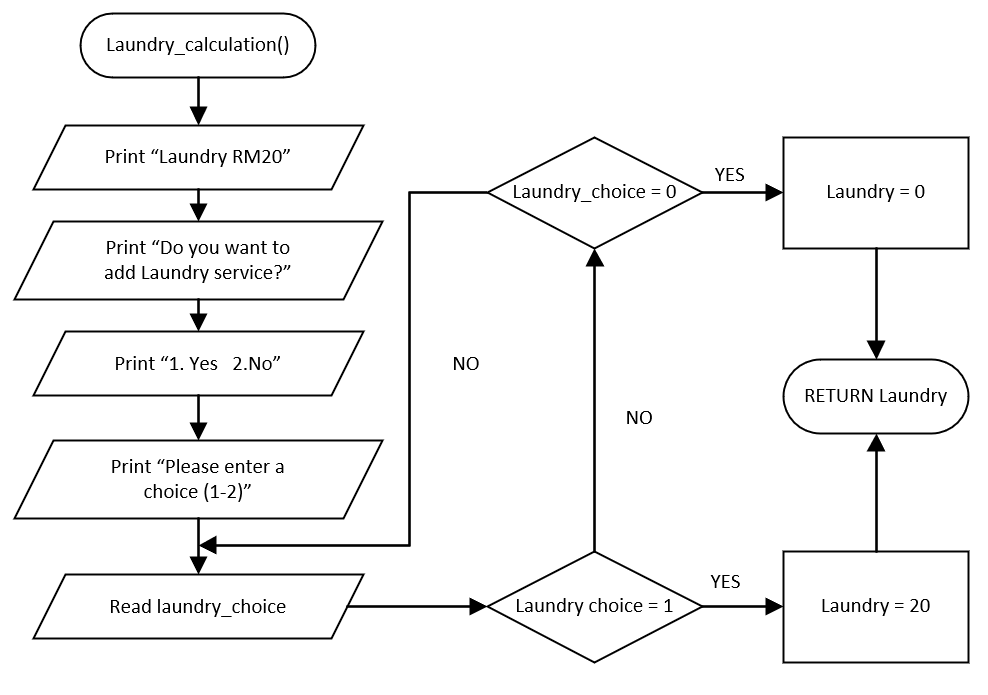
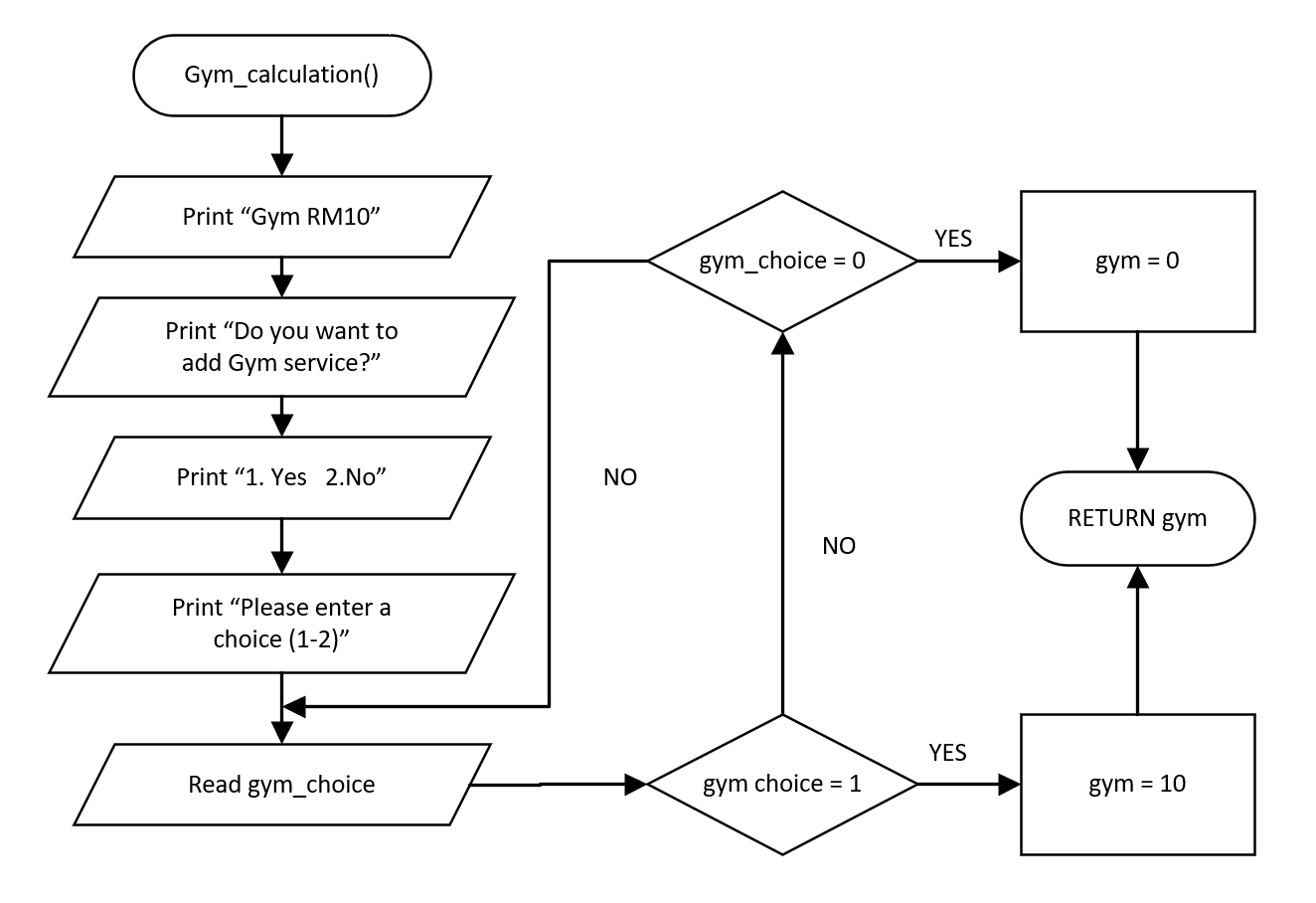


Figure 2.2.16 Gym Calculation Function

Figure 2.2.15 Laundry Calculation Function

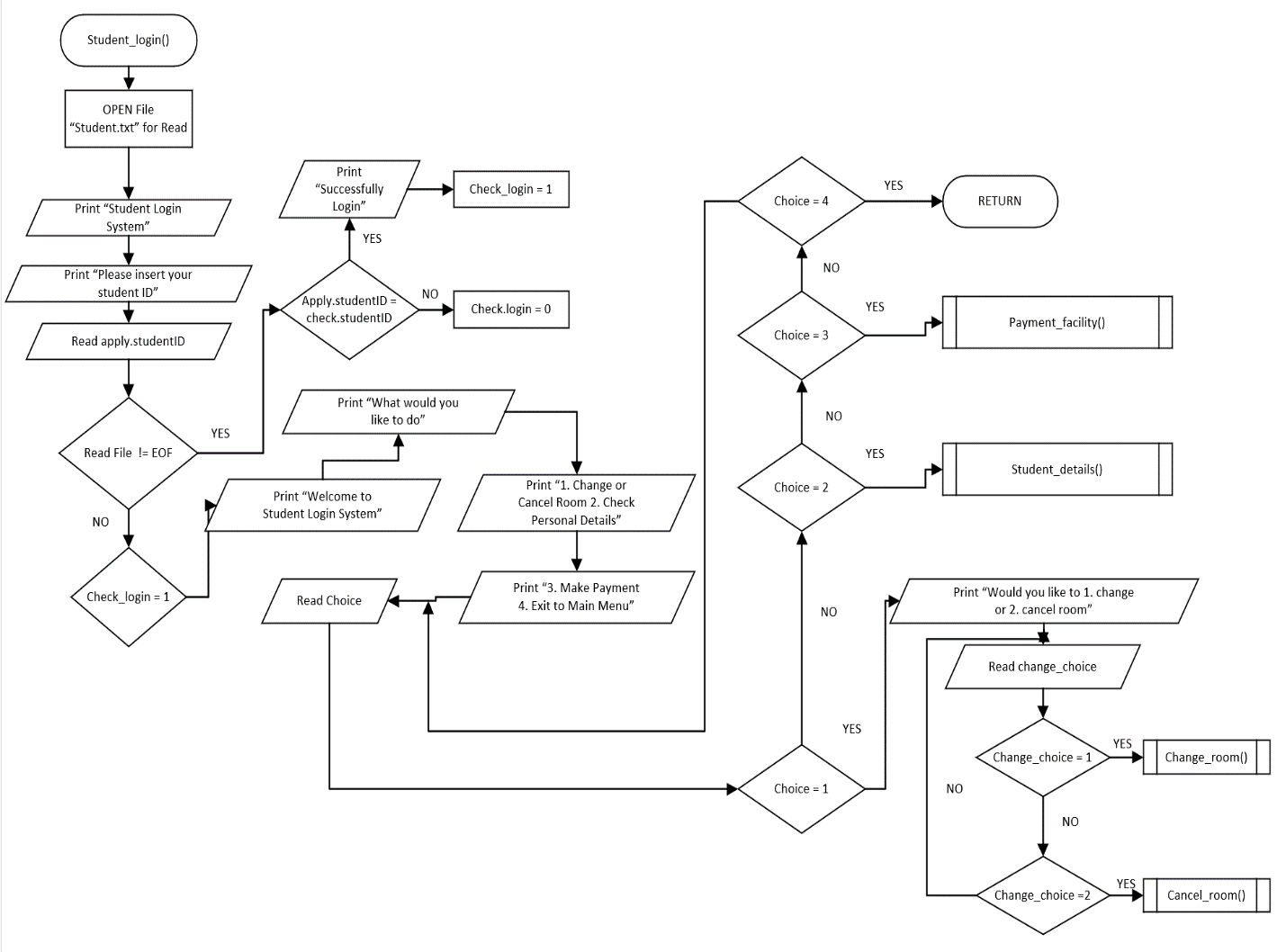
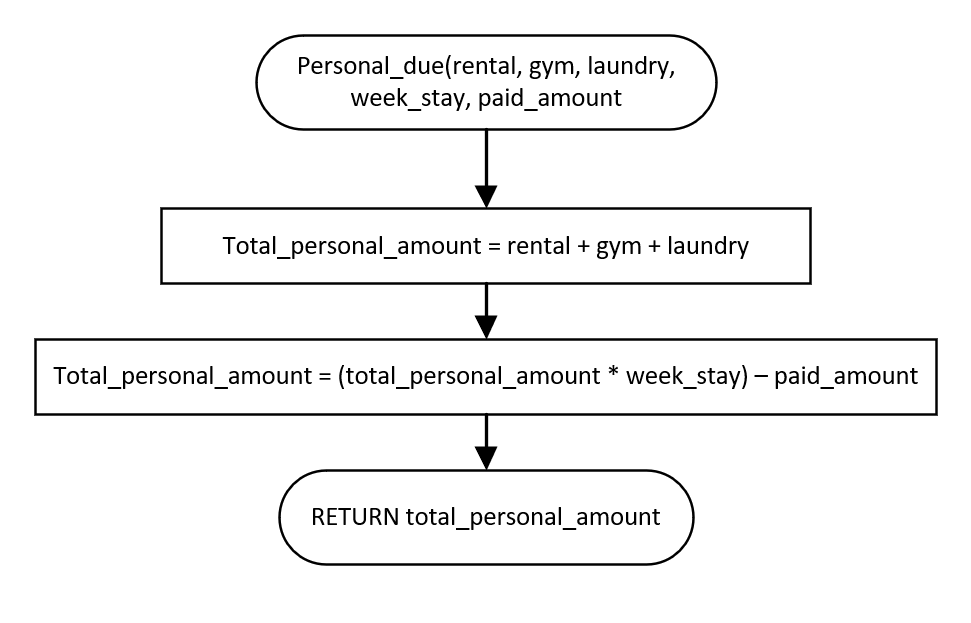
 

Figure 2.2.18 Student Login Function

Figure 2.2.17 Personal Due Function

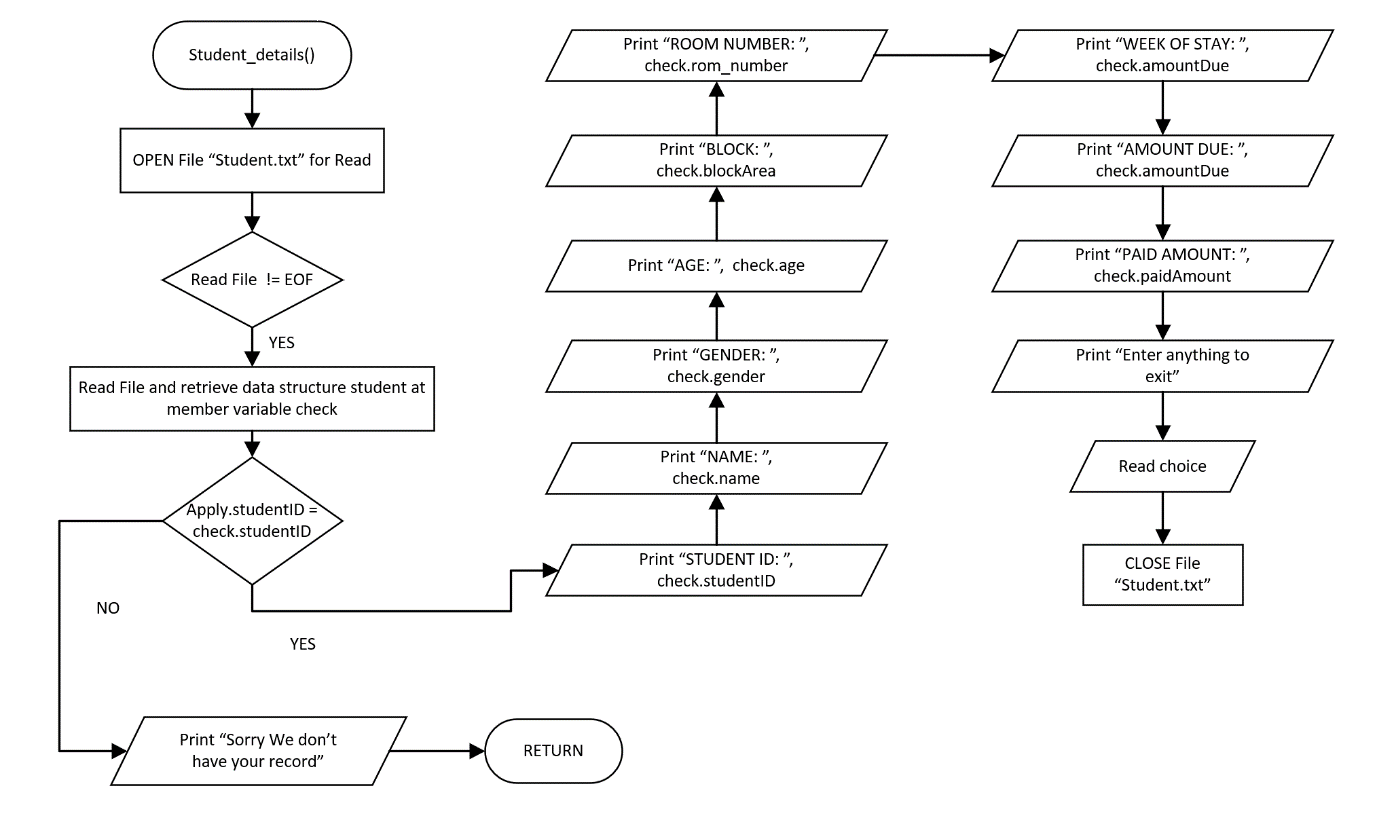


Figure 2.2.20 Student Details Function

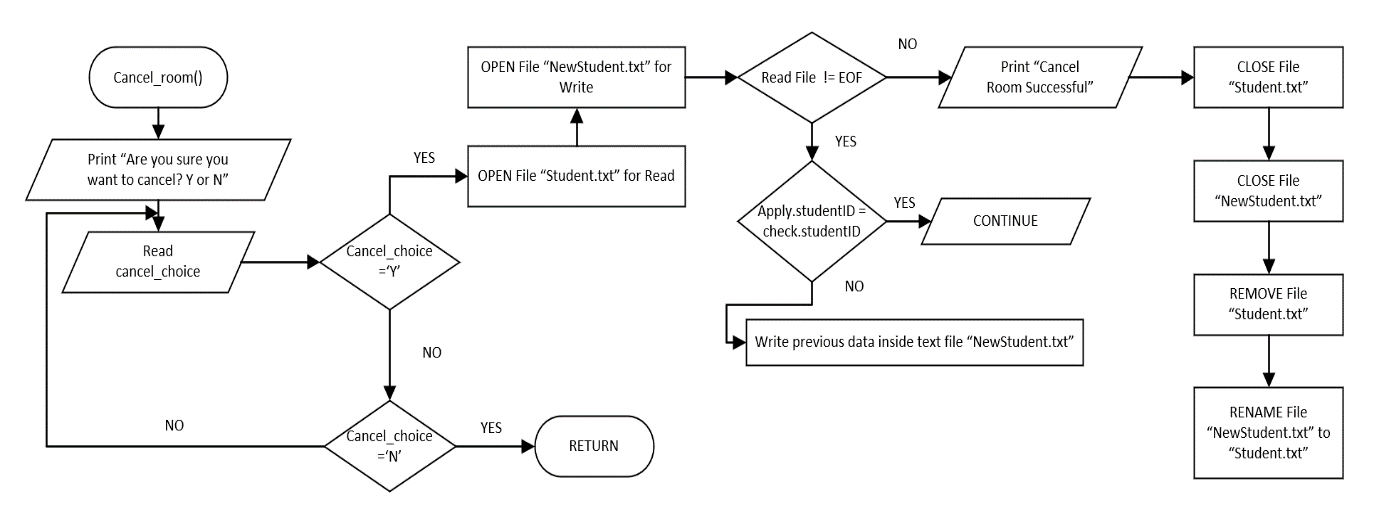


Figure 2.2.19 Cancel Room Function

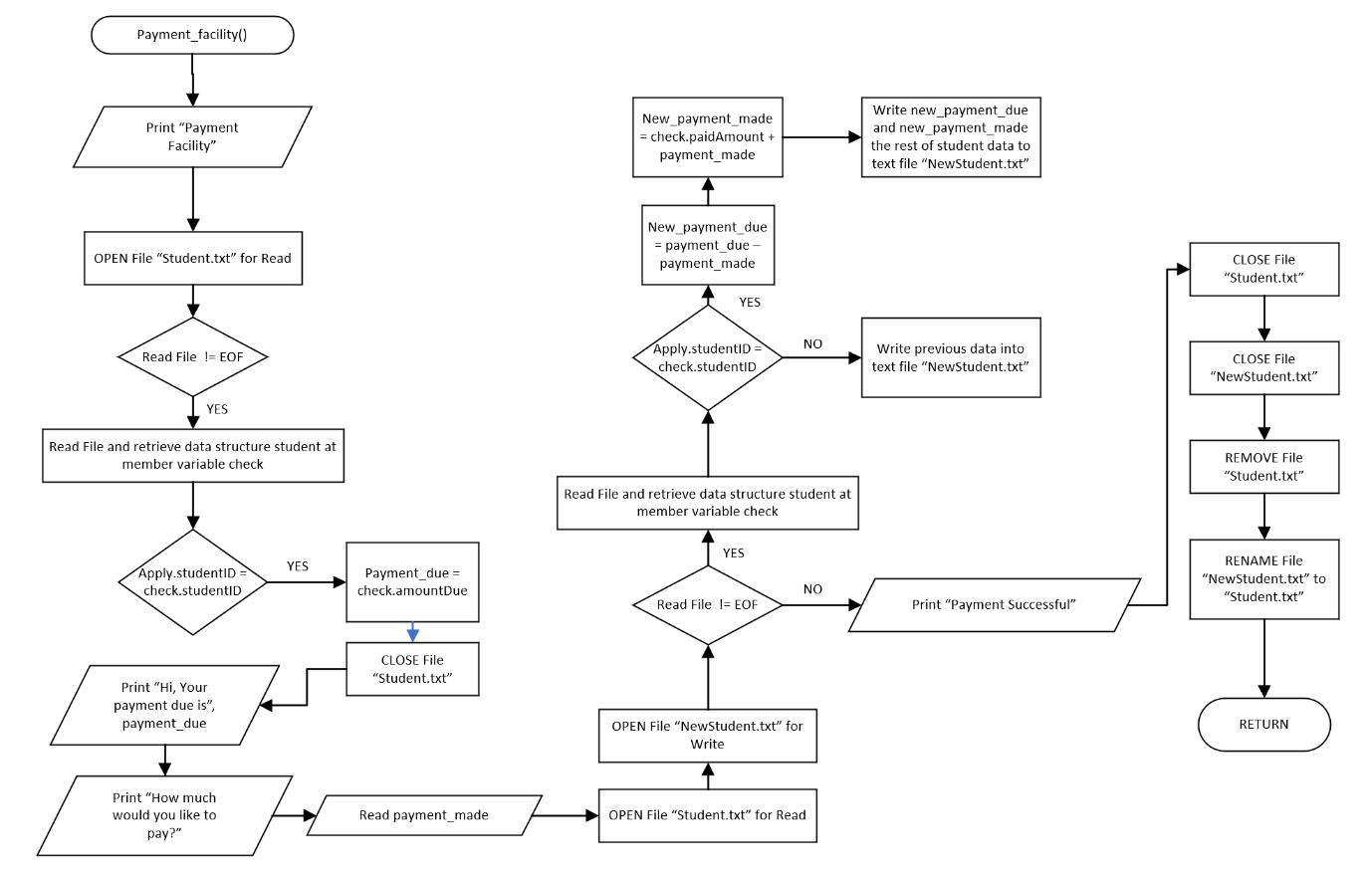
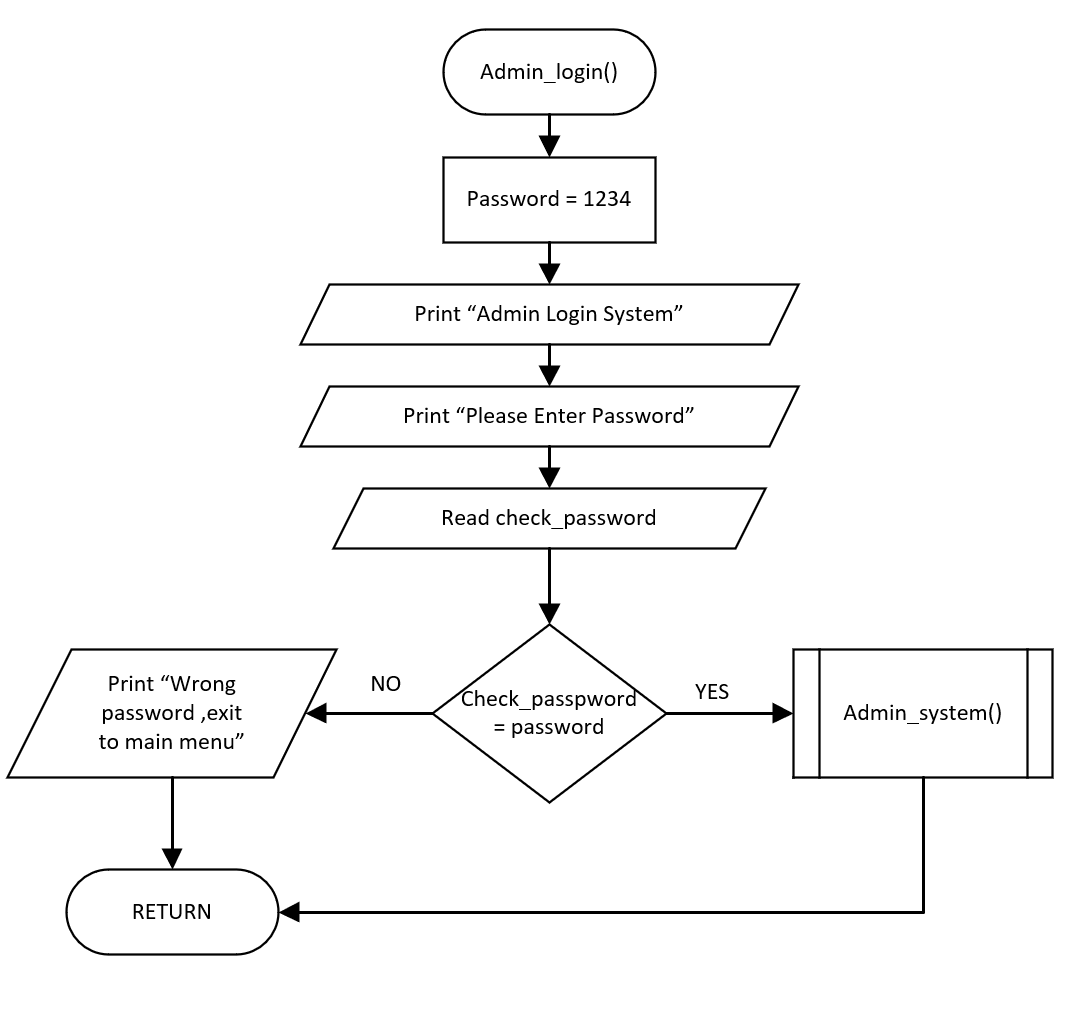


Figure 2.2.22 Admin Login Function

Figure 2.2.21 Payment Facility Function

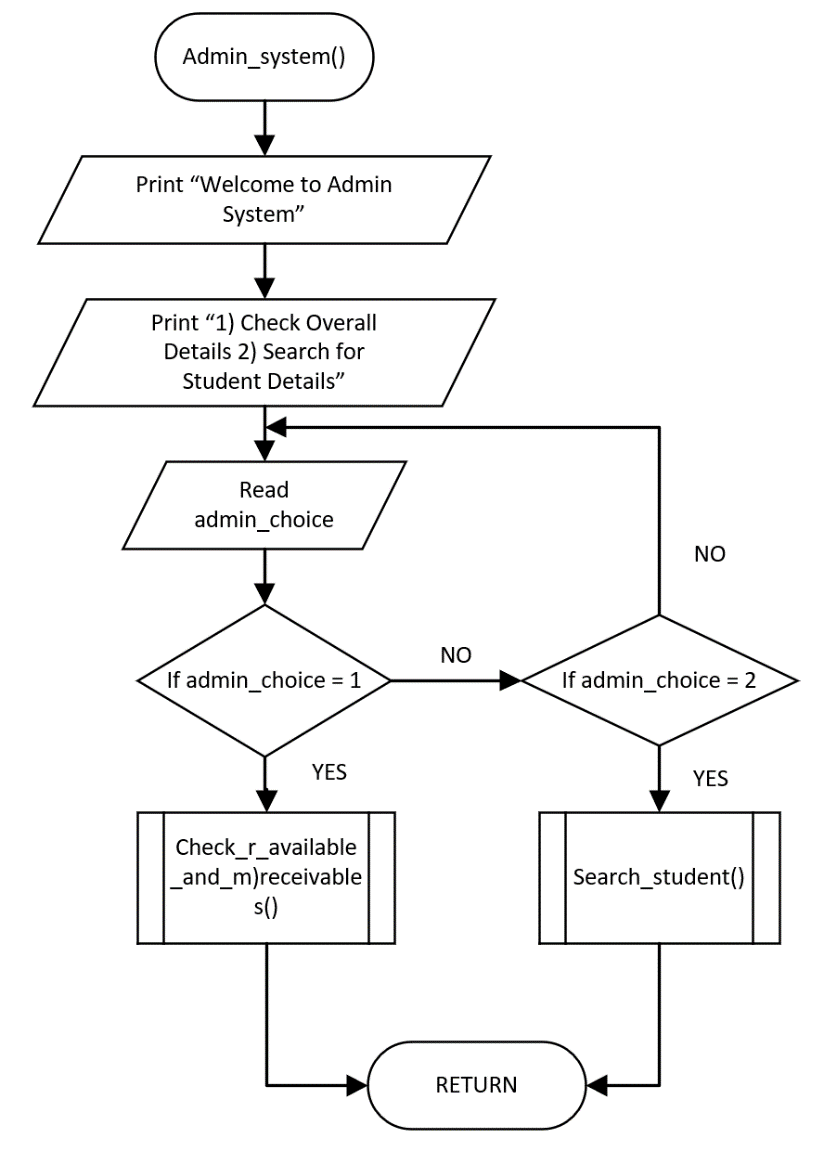


Figure 2.2.23 Admin System Function

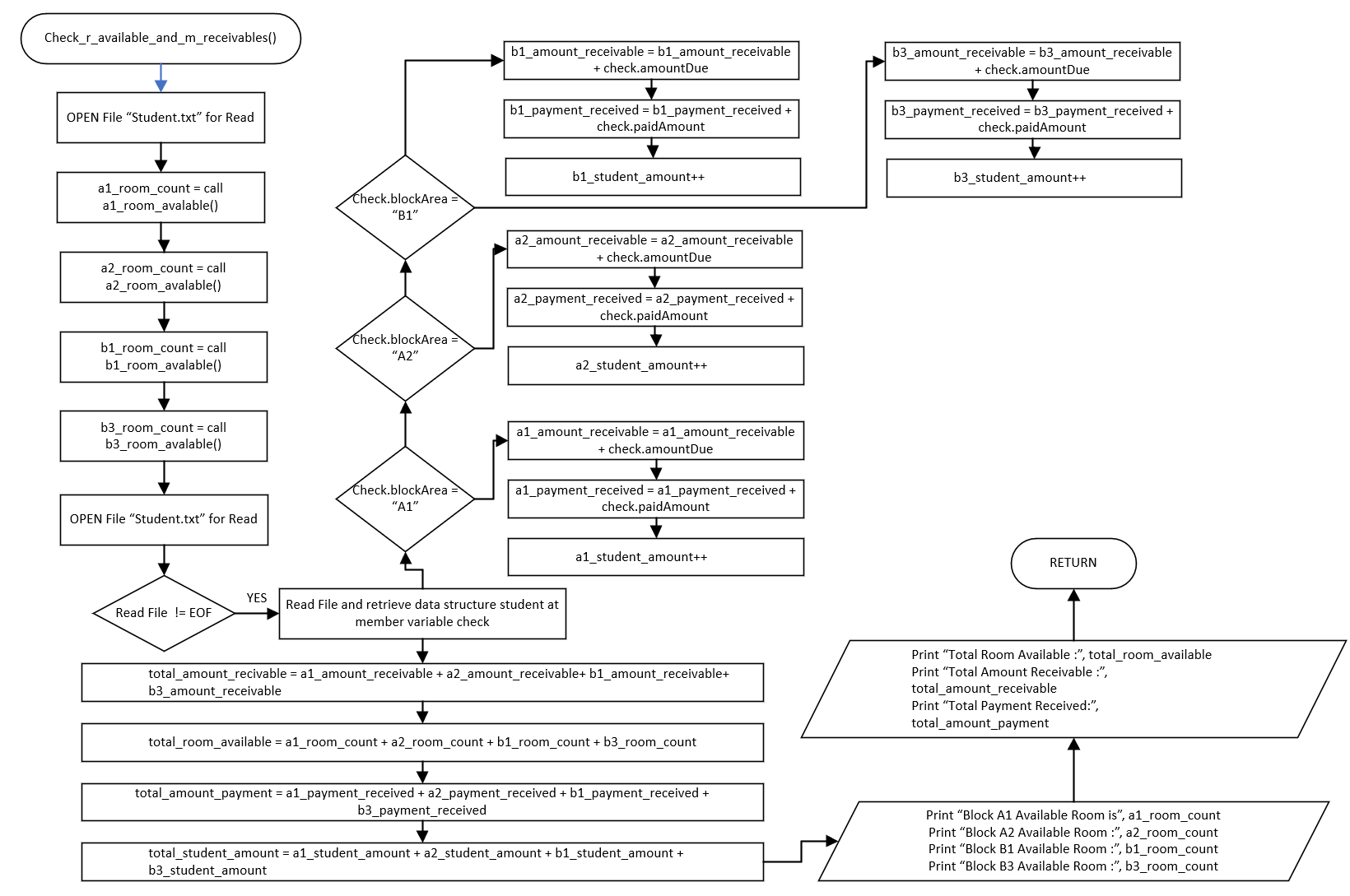


Figure 2.2.24 Check Room Availability and Money Receivable Function

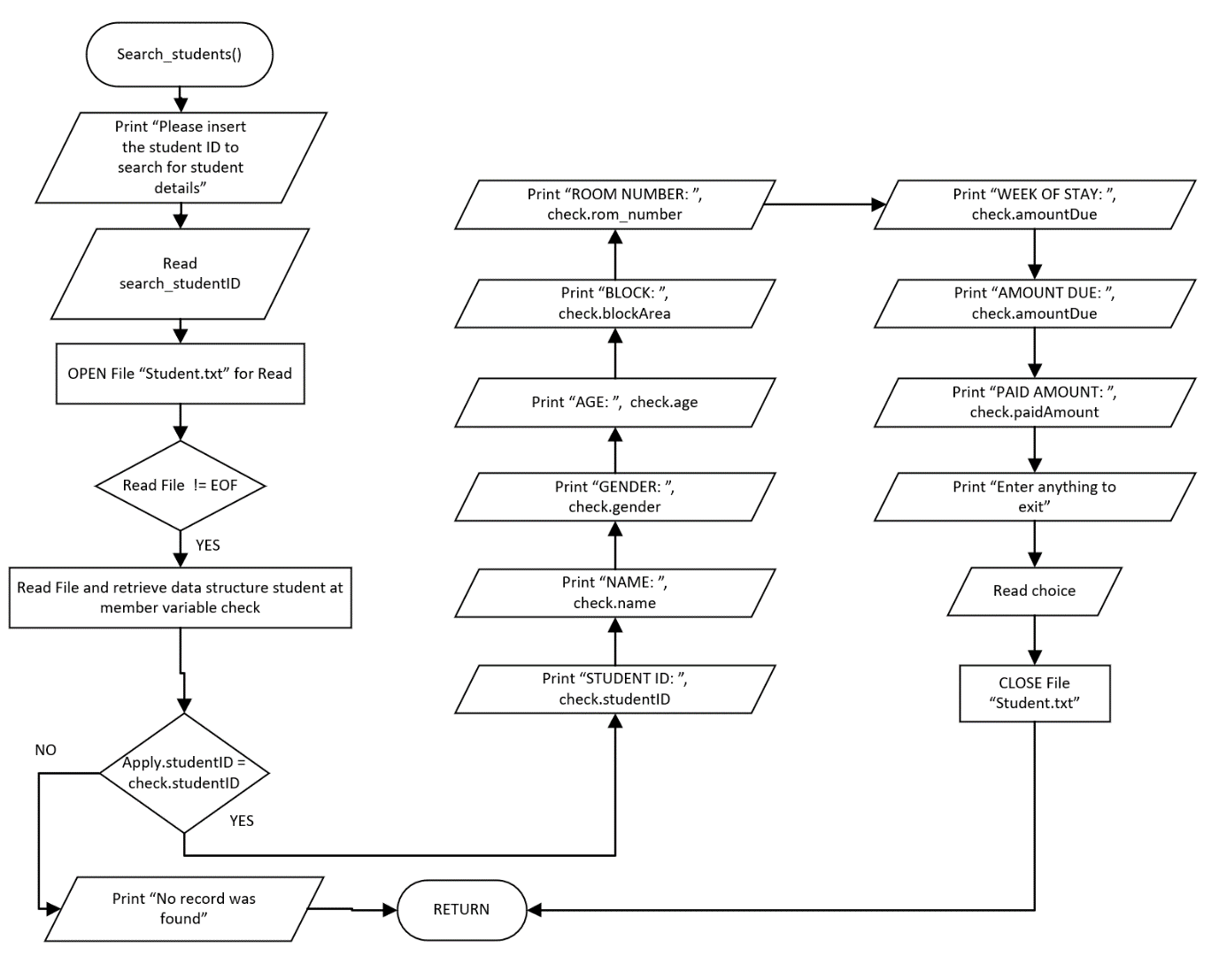


Figure 2.2.25 Search Student Function

# 3.0 | C Programming Concept

## 3.1 | Header File

Figure 3.1 Header File

A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source file. There are two types of header files, the file that the programmer writes and the files that comes with the compiler. The header file that I use is “#include <stdio.h>”, “#include <string.h>”, “#include <stdlib.h>”.

## 3.2 | Data Structures

Figure 3.2 Data Structure

Data structured that I applied inside the assignment is for student details, which is studentID, firstName, lastName, gender, identification, age, phone\_num, email, block\_area, room\_num, laundry service, gym\_service, wekk\_stay, amount\_due, paid\_amount, bed\_num. There are 2 data structure that I applied, one for apply, one for check.

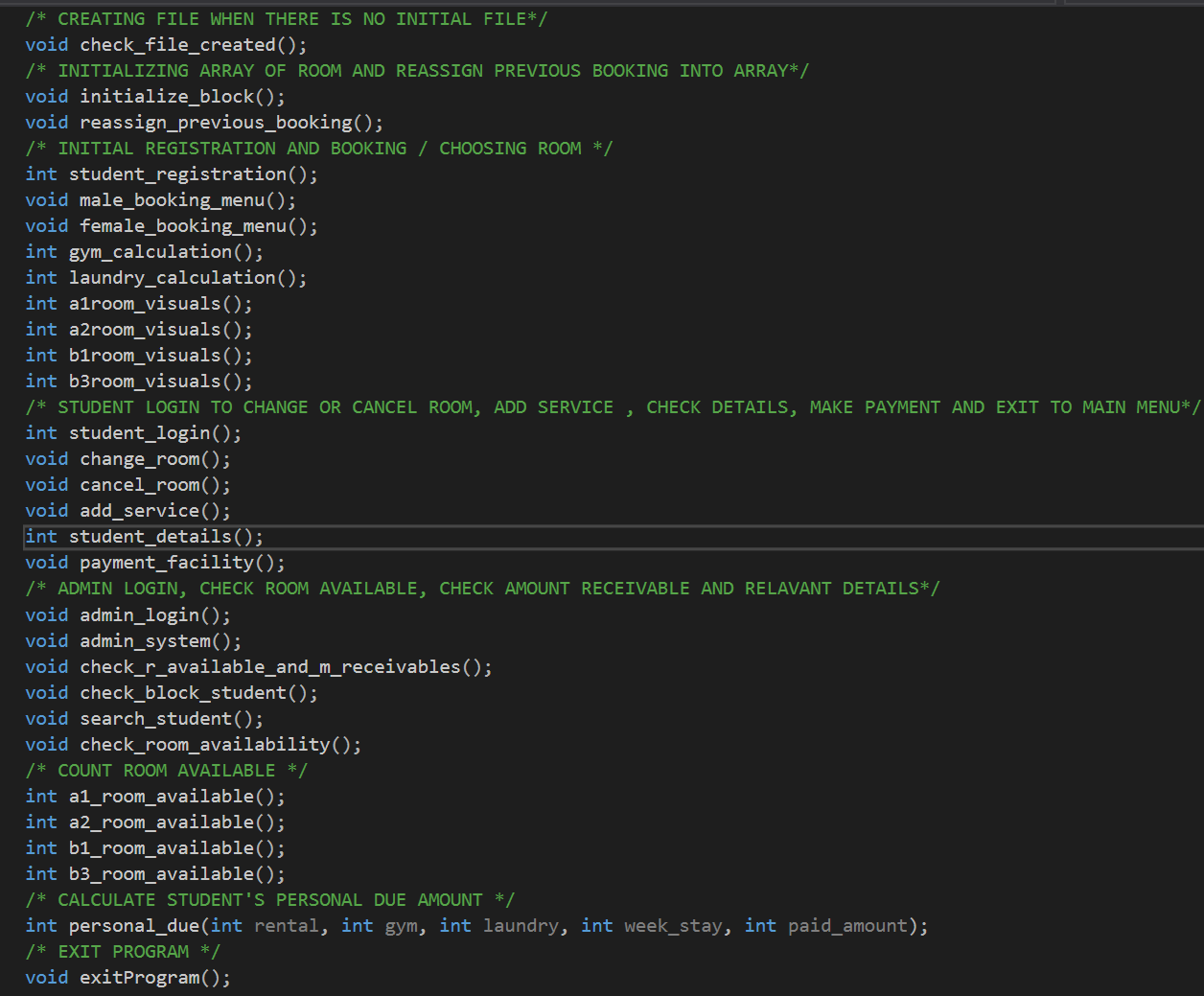
3.3 | Function

Figure 3.1 Total Function

Functions are groups of statements that perform the same programmed task. I’ve used different function grouped based on their functionality which is student registration and booking, student login for changing details and check details, admin login and so on.

## 3.4 | Displaying Text

Figure 3.4 Displaying Text

Printf function is used like shown in figure 3.4 to display text. Printf function is included in the header file in compile under “#include <stdio.h>”.

## 3.5 | Reading Value

Figure 3.5 Reading Value

Scanf function is used to store a value, as shown in figure 3.5, scanf function will request an input of “%d” which is decimal to the variable choice. Scanf function is included in the header file in compile under “#include <stdio.h>”.

## 3.6 | Assigning Value to Variable

Figure 3.6 Assigning Value to Variable

Assigning variable is used to initialize the value of the variable. Figure 3.6 assigns the value 400 into the variable rental. Before assigning value, the variable must be first declared into either int, float, string, character or so on.

## 3.7 | Comment

Figure 3.7 Comment

Comment are used in programs to let other user understand their code, giving comments of what the function does. Comment are used by simply inserting based on the format /\* (text) \*/. Figure 3.7 shows that this comment is to allow users to know it is the main function.

## 3.8 | If, Else Statement

Figure 3.8 If, Else Statement

If else statement is used to check conditions, for this instance if condition is ‘A’ and ‘1’ then it proceeds to execute the statement below and else if another condition and else the remaining conditions.

## 3.9 | Switch, Case

Figure 3.9 Switch Case Statement

Switch statement is used to test a variable against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case. In this instance, switch is checking the variable add\_choice if it is case ‘L’ or ‘l’ if proceeds to execute the statement below.

## 3.10 | While Loop

Figure 3.10 While Loop

While Loop will repeatedly execute a target statement if the set condition is true. Figure 3.10 has shown a while loop of while scanning file is not end of file its going to continuously execute until it reaches the end of the file.

## 3.11 | For Loop

Figure 3.11 For Loop

For Loop is a statement that allows programmers to write a loop that suit the needs of executing a specific number of times. Figure 3.7 will run 100 times, because there’s a for loop of 10 and a inside loop of 10. So, it will run 10 times x 10 times which is 100 times.

## 3.12 | Open / Close File

Figure 3.12 Open and Close File

To read, write and append information or data inside the file, file must be open and initialize for the usage either read, write or append. After opening the file, we must close the file after we finish using the file for read, write or append.

## 3.13 | Declaration Variable

Figure 3.13 Declaration Variable

Variable can be declared as int, float, char and so on. Variable should be declared in a name that is easily understandable.

## 3.14 | Read File

Figure 3.14 Read File

When files are opened, we will need to specify the usage. In figure 3.14, “Student.txt” is opened for reading, so it will read the information or data inside the text file and no modification of the information is allowed while reading file.

## 3.15 | Write File

Figure 3.15 Write File

If file is specified to write, which is what is shown on figure 3.15. The program will overwrite or write on the “NewStudent.txt” file. If there is stored info inside “NewStudent.txt”, it is going to overwrite the whole file.

## 3.16 | Append File

Figure 3.16 Append File

If file is specified to append, which is what is shown on figure 3.16. The program will append and add information inside “Student.txt”, while not overwriting the old info that is inside the file. So, the new information will be added inside the file and the old info will remain.3.17 | Strcpy Function

Figure 3.17 Strcpy Function

The function strcpy is a built-in function included in the header file “#include <string.h>” that is included in the compiler. Strcpy is a function that copies an array of character which is also string inside the variable. In figure 3.17 the string or character array of “B3” will be copied inside the variable apply. block\_area. (Programiz, n.d.)

## 3.18 | Strcmp Function

Figure 3.18 Strcmp Function

The function strcmp is a built-in function included in the header file “#include <string.h>” that is included in the compiler. Strcmp is a function that compares 2 string variables. In figure 3.18, the program compares string apply. studentID with check. studentID. If there is no difference, the output will be 0. (Tutorialspoint, n.d.)

## 3.19 | Go to

Figure 3.19 Go to Function

The function goto is function that allows program to go back a certain point of the program. For this instance, in figure 3.19, cancel\_menu: is created as a location, and if in a certain point I want to go back to the location and run the program from the location again, goto function will be used as shown in figure 3.19. (Singh, 2015)

## 3.20 | Array

Figure 3.20 Array

Array is a collection of elements that could store multiple data. As shown in figure 3.20. block\_a1 and block\_b1 is going to have 100 elements, and block\_a2, block\_b3 are having 2 and 3 elements inside the 100 elements as represent to beds in the room.

## 3.21 | System (“CLS”)

Figure 3.21 System ("CLS")

 System(“CLS”) function is used to clear the printed text when running the program.

## 3.22 | Sleep Function

Figure 3.22 Sleep Function

 Sleep Function is called to make a delay to execute the next line of program. As shown above on figure 3.22, the syntax is \_sleep() and the number inside the bracket represents milliseconds. So, Figure 3.22 will delay 2000 millisecond which is also 2 second.

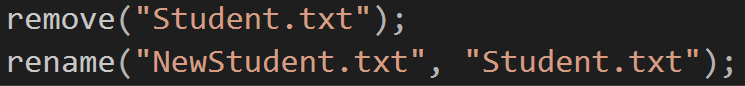
## 3.23 | Strupr Function

Figure 3.23 Strupr Function

 Strupr function is used to make all string into upper case. Figure 3.23 converts all string inside check. studentID to upper case letters.

## 3.24 | Remove / Rename File

Figure 3.24 Remove / Rename File

 Remove function is used to delete a text file and rename function is used to rename file. As shown above in figure 3.24, the program will delete “Student.txt”, rename “NewStudent.txt” to “Student.txt”.

## 3.25 | Fscanf Function

Figure 23.25 Fscanf Function

Function fscanf reads formatted input from a string and will then assign the scanned value into the variable that is desired. (Tutorialspoint, n.d.)

# 4.0 | Additional Features

## 4.1 | Admin Login

Figure 4.1.2 | Admin Login Sample Output

Figure 4.1.1 Admin Login Source Code

Figure 4.1.1 shows the source code of the password which is “AbcD1234”, if the password is correct, the program will proceed to redirect users to the admin system as shown in Figure 4.1.2.

## 4.2 | Choosing Room

Figure 4.2.2 Choose Room Sample Output

Figure 4.2.1 Choose Room Source Code

Figure 4.2.1 shows the source code of choosing room. As shown in figure 4.2.2, the program will display all the available room as 0 and chosen room will be shown as 1. User can choose their desire room number if the room is available.

## 4.3 | Change Student Personal Details

Figure 4.3.2 Change Personal Details Sample Output

Figure 4.3.1 Change Student Personal Details Source Code

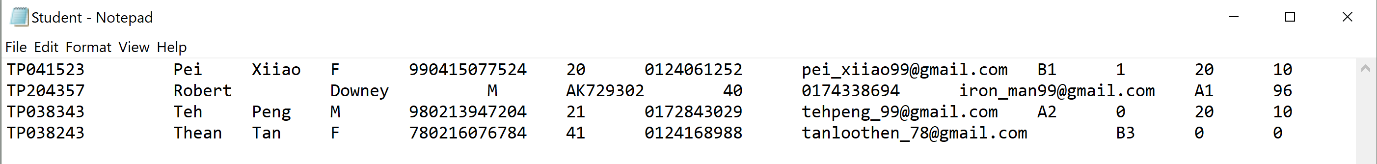
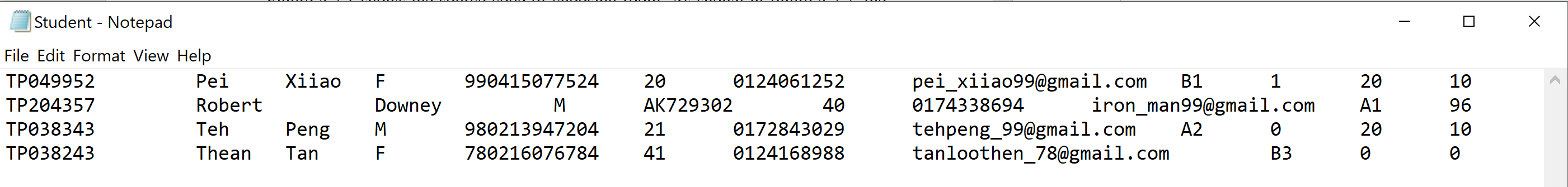


Figure 4.3.4 File Information After Change of Details

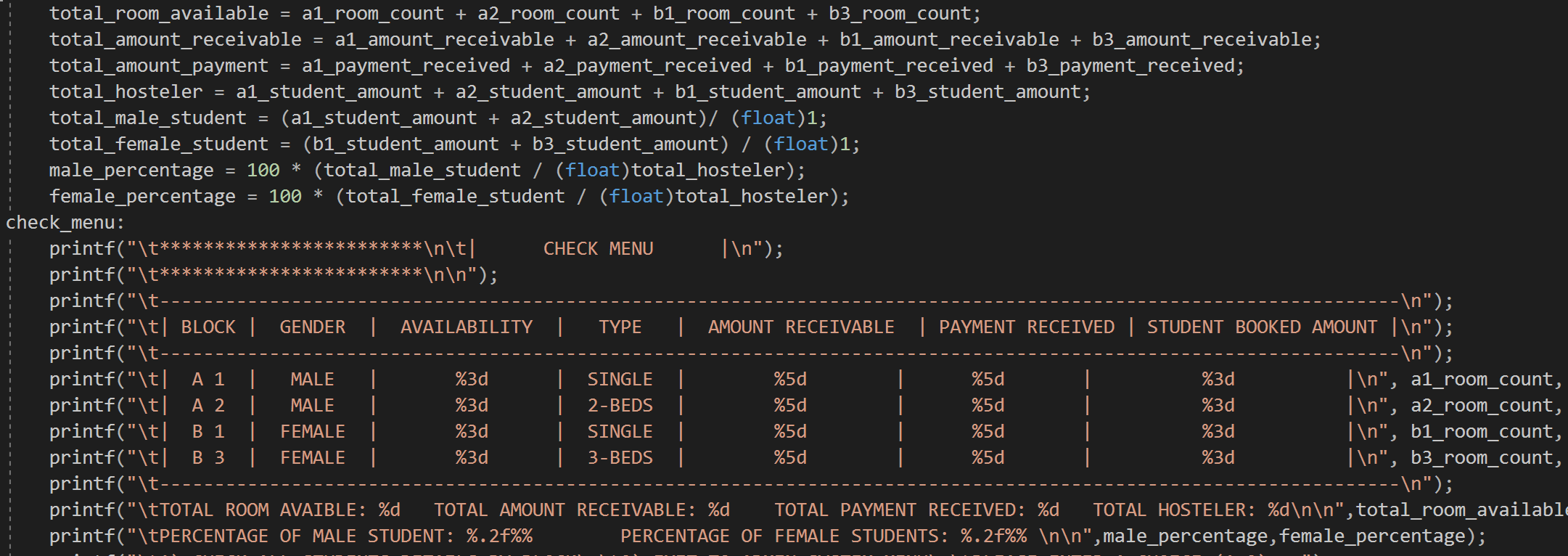
Figure 4.3.3 File Information Before Change of Details

Figure 4.3.1 shows the source code of allowing students to change their personal details if they found that details are incorrect. As shown in Figure 4.3.2, once you enter y for making changes, the program will again request for another input from the user asking which details they would want to change from 1) student id to 7) email. For this instance, user input 1 which leads to changing their student ID, after input TP049952, as shown above, figure 4.3.3 is the one before the change. After the input, the program will then write the change inside the text file as shown in figure 4.3.4. So, the Student ID will change from TP041523 to TP049952 and it will be written inside the text file for future use.

## 4.4 | Show Percentage of Male and Female Hosteler

Figure 4.4.2 Percentage of Male & Female Hosteler Sample Output

Figure 4.4.1 Calculation & Showing of Male & Female Hosteler Percentage Source Code

 Figure 4.4.1 shows the calculation and the source code of the check menu and as shown in figure 4.4.2, the program is going to show all the necessary details like room availability, total payment receivable, payment received and especially the percentage of male and female students as shown in figure 4.4.2, there is 61.9 % male students and 38.1 % female students.

# 5.0 | Test Specification Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function** | **Input** | **Expected Output** | **Given Input** | **Output** | **Results** |
| Main Manu | 1 | Enter to Student Registration Menu | 1 | Enter Registration Menu | Pass |
| 2 | Enter to Student Login Menu | 2 | Enter Student Login Menu | Pass |
| 3 | Enter Admin Login Menu | 3 | Enter Admin Login Menu | Pass |
| 3 | Redirect to Main Menu | Fail |
| 4 | Exit Program | 4 | Exit Program | Pass |
| 4 | Redirect to Main Menu | Fail |
| Student Registration Menu | Registered Student ID | Redirect Back to Main Menu | Registered Student ID | Proceeds to Register | Fail |
| Proceeds to Register | Registered Student ID | Redirect to Main Menu | Pass |
| New Student ID | Proceeds to Register | New Student ID | Redirect to Main Menu | Fail |
| New Student ID | Proceeds to Register | Pass |
| Registered Student ID | Proceeds to Register | Fail |
| Student Login Menu | Registered Student ID | Proceeds to Student Login System | Registered Student ID | Proceeds to Student Login System | Pass |
|  | Registered Student ID | Redirect to Main Menu | Fail |
| New Student ID | Redirect to Main Menu | New Student ID | Redirect to Main Menu | Pass |
| New Student ID | Proceeds to Student Login System | Fail |
| Student Login System | 1 | Enter choice Change or Cancel Room | 1 | Returned to Student Login Menu | Fail |
| 1 | Enter choice Change or Cancel Room | Pass |
| 1 | Shows Booking Menu | Fail |
| 1 | Redirect to Main Menu | Fail |
| 2 | Shows Personal Details and ask for user if they want to change | 2 | Shows Personal Details and ask for user if they want to change | Pass |
| 2 | Redirect to Main Menu | Fail |
| 2 | Redirect to Admin Login Menu | Fail |
| 3 | Direct to Payment Facility to make payment | 3 | Direct to Payment Facility to make payment | Pass |
| 3 | Remains in Student Login Menu | Fail |
| 4 | Exit to Main Menu | 4 | Exit Program | Fail |
| 4 | Exit to Main Menu | Pass |
| 4 | Redirects to Student Login Menu | Fail |
| Admin Login Menu | 1 | Ask for password Input | 1 | Ask for password Input | Pass |
| 1 | Redirects to Admin System | Fail |
| 1 | Exits to Main Menu | Fail |
| Password | Redirects to Admin System | Password | Login into Admin System | Pass |
| Password | States wrong password and request for password again | Fail |
| 2 | Exit to Main Menu | 2 | Remain in Admin Login Menu | Fail |
| 2 | Exit to Main Menu | Pass |
| Admin System | 1 | Shows overall details like total amount receivable and total students | 1 | No record found, and only displays 0 | Fail |
| 1 | Redirect to Student Search Engine | Fail |
| 1 | Shows overall details like total amount receivable and total students | Pass |
| 2 | Redirects to Student Search Engine | 2 | Shows overall details, total amount receivable and total amount received | Fail |
| 2 | Redirects to Student Search Engine | Pass |
| 3 | Ask user to choose block to view availability | 3 | Shows total percentage of male and female student | Fail |
| 3 | Redirect to Main Menu | Fail |
| 3 | Exit Program | Fail |
| 3 | Ask user to choose block to view availability | Pass |
| 4 | Exits to Main Menu | 4 | Exit to Main Menu | Pass |
|  | 4 | Exit Program | Fail |
| Student Registration System | New Student ID | Proceed to Enter personal Details | New Student ID | Redirects to Main Menu because record was found | Fail |
|  | New Student ID | Proceed to Enter personal Details | Pass |
| Registered Student ID | Redirects to Main Menu | Registered Student ID | Redirects to Main Menu | Pass |
|  | Registered Student ID | Proceeds to Enter personal details | Fail |
| Male / Female Booking Menu Choose Room | 1 to 99 | Booking Successful if room is available, unsuccessful booking if room is booked by other students | 7 | Successful Booking, and room is available | Pass |
| 8 | Successful Booking, and room is available | Pass |
| 9 | Successful Booking, and room is available | Pass |
| 11 | Unsuccessful Booking because room is booked by other student | Pass |
| Male / Female Booking Menu Choose Bed | L, l | Left Bed is available successfully booked bed | L | Booking successful although it is booked | Fail |
| l | Successfully booked available bed | Pass |
| M, m | Middle Bed is available successfully booked bed | M | Booking fail although bed is available | Fail |
| m | Successfully booked available bed | Pass |
| R, r | Right Bed is available and successfully booked bed | R | Successfully booked available bed | Pass |

# 6.0 | Sample Output:

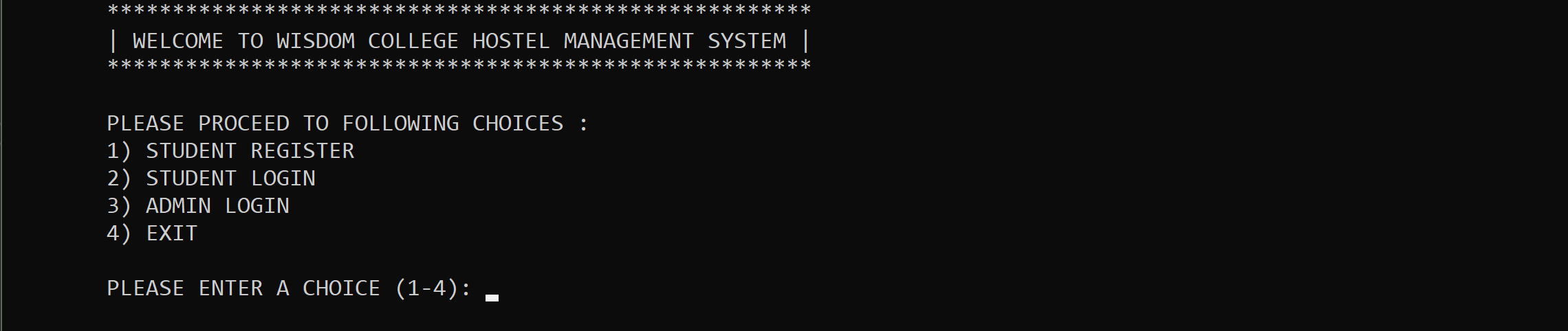
6.1 | Hostel Management System Sample Output:

Figure 6.1: Hostel Management System Main Menu

When program is first executed, the program will display the hostel management system main menu with 4 choices which is 1. Student register, 2. Student login, 3. Admin login, 4. Exit.

## 6.2 | Student Register Sample Output

Figure6.2.2: If Enter Choice 1

Figure 6.2.1 If Enter Choice 1

After the input 1, the program will then direct students to the student registration system, for them to proceed register to book a hostel room. The program will then ask the user to input the student ID with the format TP000000.

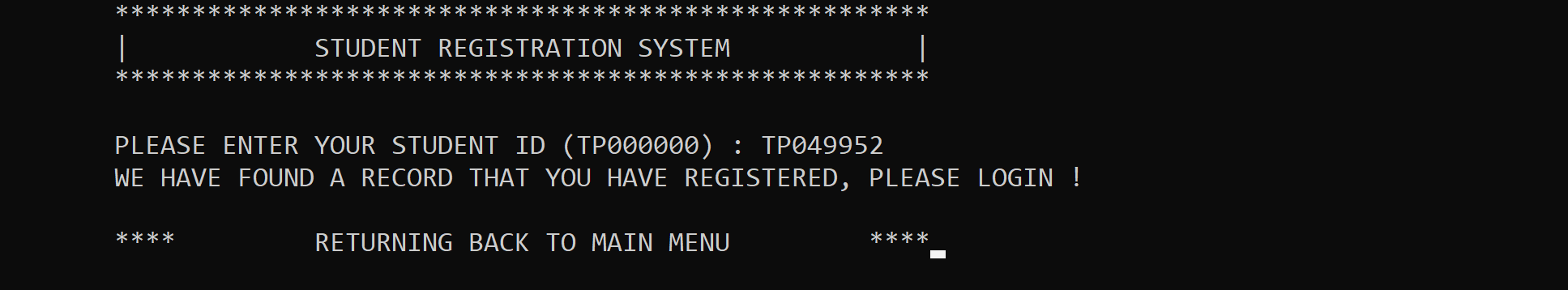
6.2.1 | Record Was Found Sample Output

Figure 3.2.1.1 Found a record of register

After input the student ID, if the student ID was found registered, the program will then output “WE HAVE FOUND A RECORD THAT YOU HAVE REGISTERED, PLEASE LOGIN!”, and the program will redirect the users back to the Hostel Management Main Menu. 6.2.2 | Successful Register Sample Output

Figure 6.2.2.1 Successful Register

If the student ID was newly registered, the program will then ask for user to input their relevant details for this instance first name, last name, gender, identification, age, phone number and email address. The program will then redirect users to the booking menu.

## 6.3 | Booking Menu

### 6.3.1 | Male Booking Menu

Figure 6.3.1.1: Male Booking Menu

After successfully registered, if student is a male, program will redirect to the male booking menu as shown in the figure above, in a clear glance the room type, block, rentals, services and room availability. Program will then request an input of 1 to 4 from user.

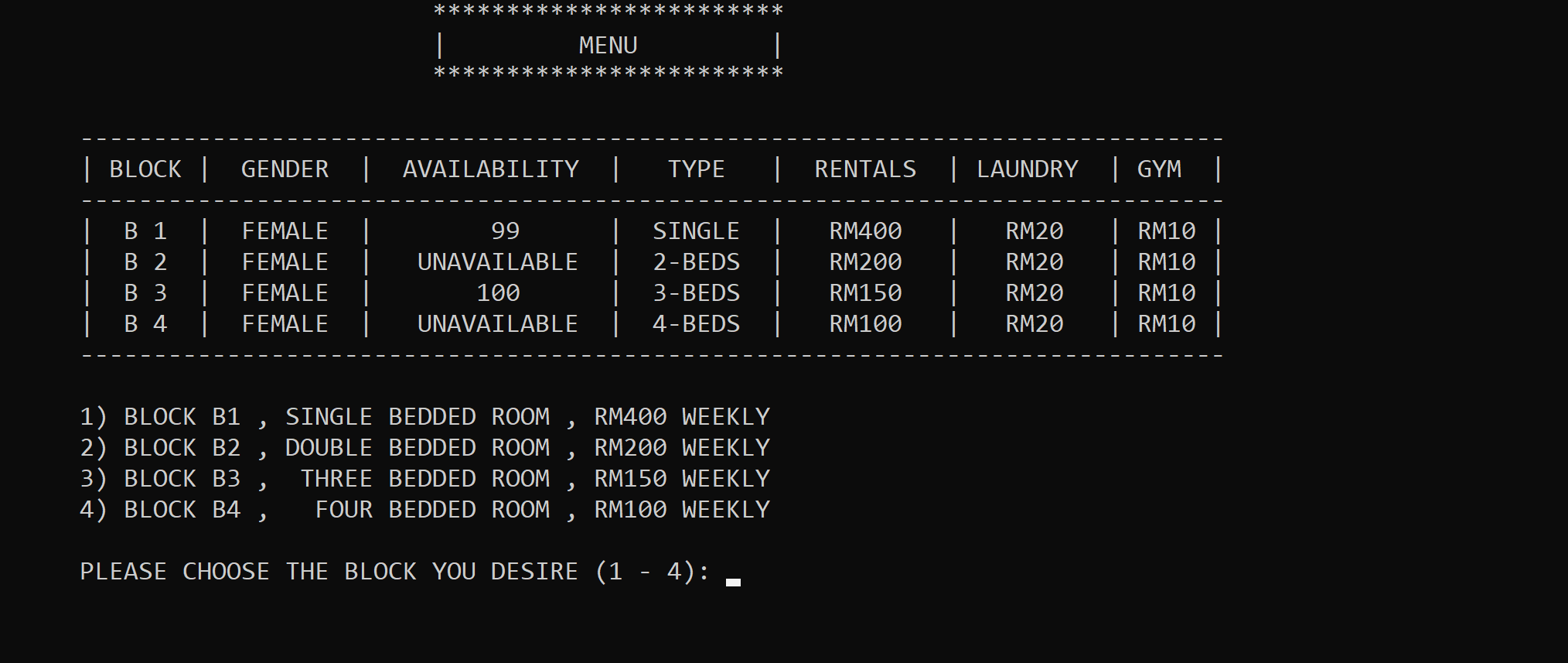
6.3.2 | Female Booking Menu

Figure 6.3.2.1 Female Booking Menu

After successfully registered, if student is a female, program will redirect to the female booking menu as shown in the figure above, in a clear glance the room type, block, rentals, services and room availability. Program will then request an input of 1 to 4 from user.

## 6.4 | Booking Room

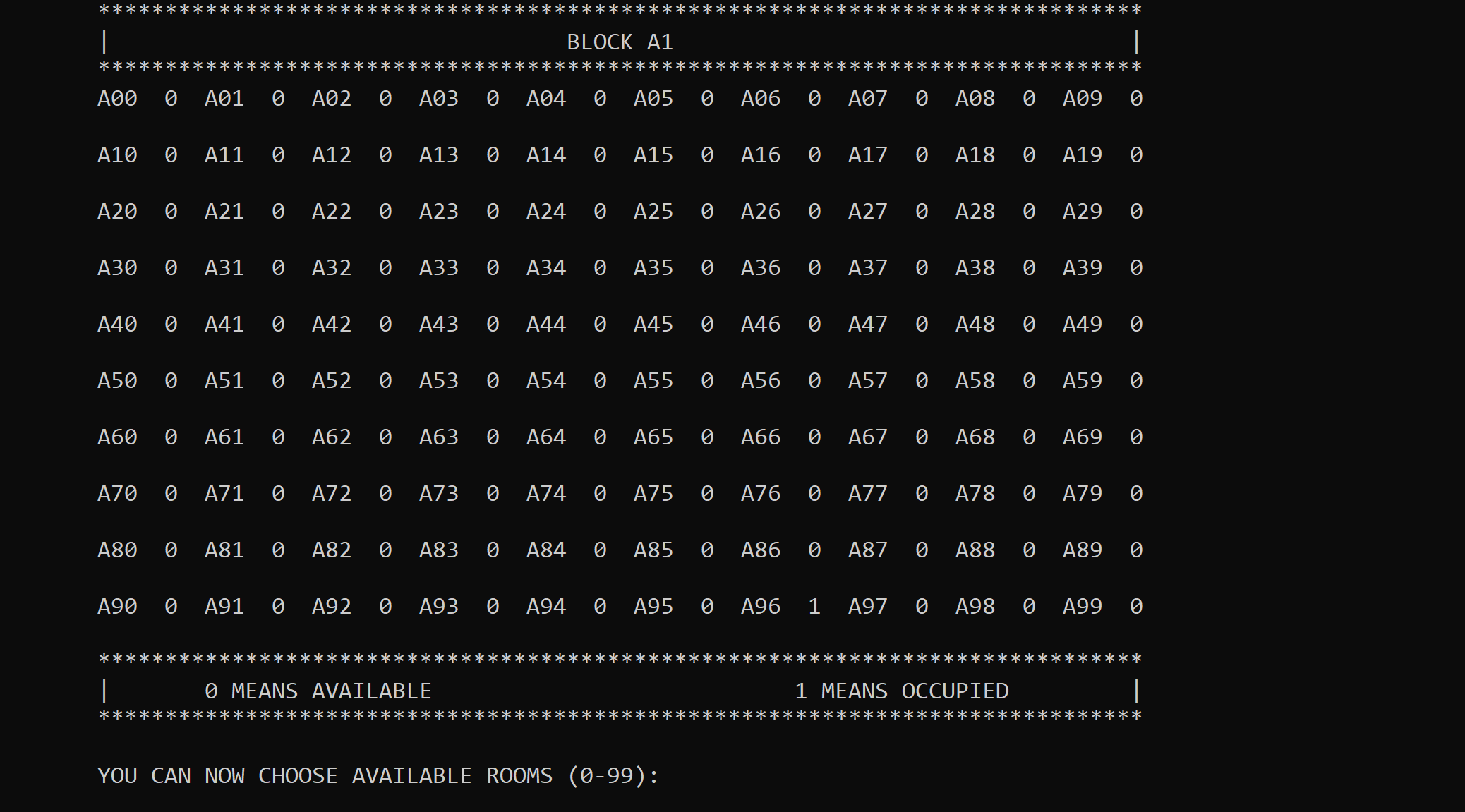
6.4.1 | Male Block A1 Booking

Figure 6.4.1.1: Block A1 Booking

If user chooses Block A1 by input 1, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it’s occupied by another student. The program will then request for an input for the room they choose.

### 6.4.2 | Male Block A2 Booking

Figure 6.4.2.1 Block A2 Booking

If user chooses Block A2 by input 2, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it’s occupied by another student. The program will then request for an input for the room they choose. As shown in the male booking menu, A2 is a double bedroom. So as figure shown above there is two 0s for users to choose. The user will be asked for choosing a room number first and later program will request user to pick the left bed or right bed.

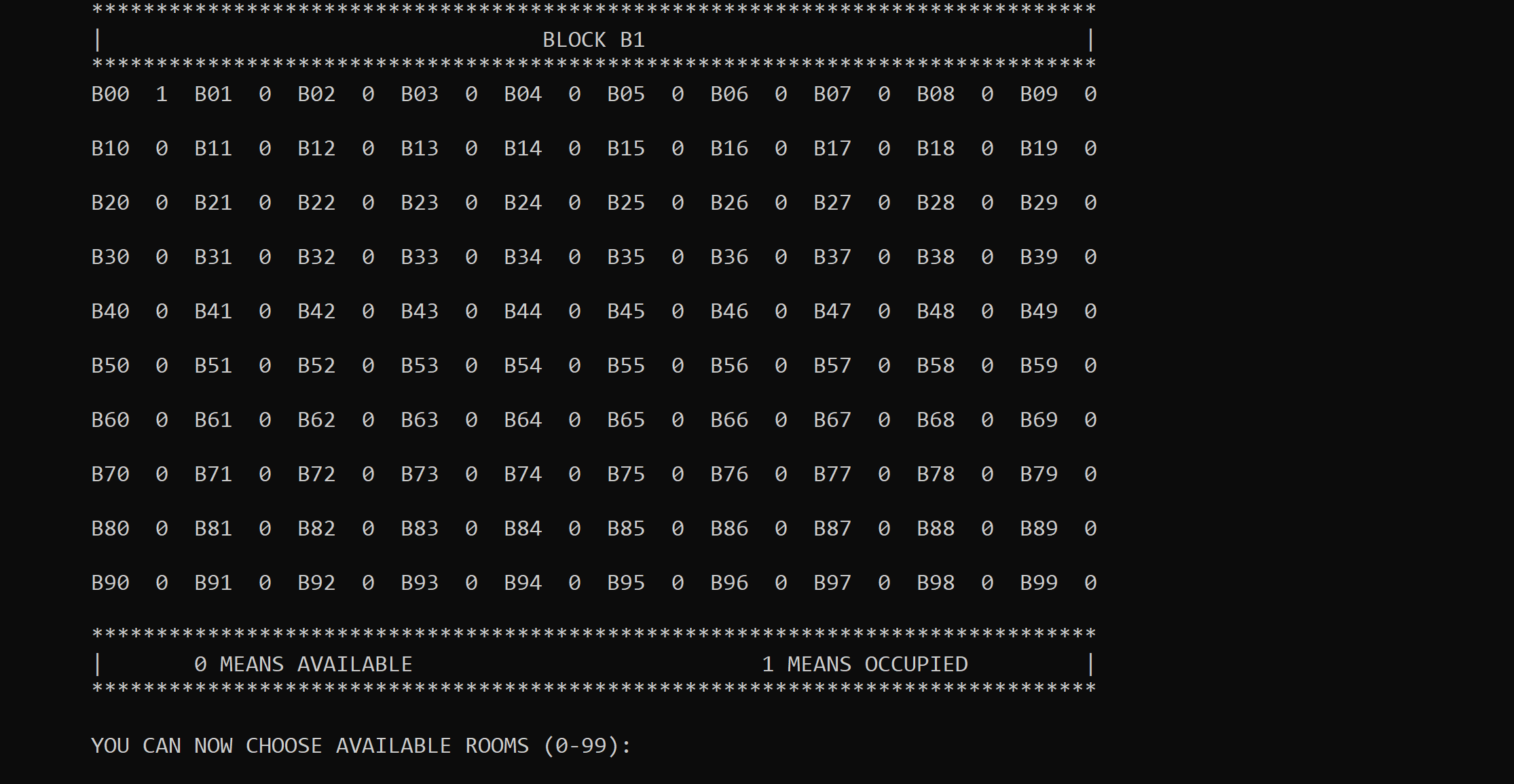
6.4.3 | Female Block B1 Booking

Figure 6.4.3.1 Block B1 Booking

If user chooses Block B1 by input 1, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it’s occupied by another student. The program will then request for an input for the room they choose.

### 6.4.4 | Female Block B3 Booking

Figure 6.4.4.1 Block B3 Booking

If user chooses Block B3 by input 3, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it’s occupied by another student. The program will then request for an input for the room they choose. As shown in the male booking menu, A2 is a double bedroom. So as figure shown above there is two 0s for users to choose. The user will be asked for choosing a room number first and later program will request user to pick the left bed or right bed.

### 6.4.5 | Unavailable Block Bookings

Figure 6.4.5.2 Unavailable Block in Female Booking Menu

Figure 6.4.5.1 Unavailable Block in Male Booking Menu

Currently, Block A3 & Block A4 in Male’s booking menu is now under construction. Block B2 & B4 in Female’s booking menu is now under construction. If user inputs either A3, A4, B2 or B4. The program will state that the block is now unavailable and will request another valid input from the user to repick their blocks again.

## 6.5 | Choosing Room Number

### 6.5.1 | Room Number Chosen is Occupied

Figure 6.5.1.1 Room is Occupied

If user inputs a room number that is occupied, in this instance is A96, the program will that output “SORRY, THIS ROOM HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER ROOM.” and request the input again from the user.

### 6.5.2 | Room Number Chosen is Available

Figure 6.5.2.1 Room is Available

If user inputs a room number that is available, the program will then redirect the user to the laundry subscription menu.

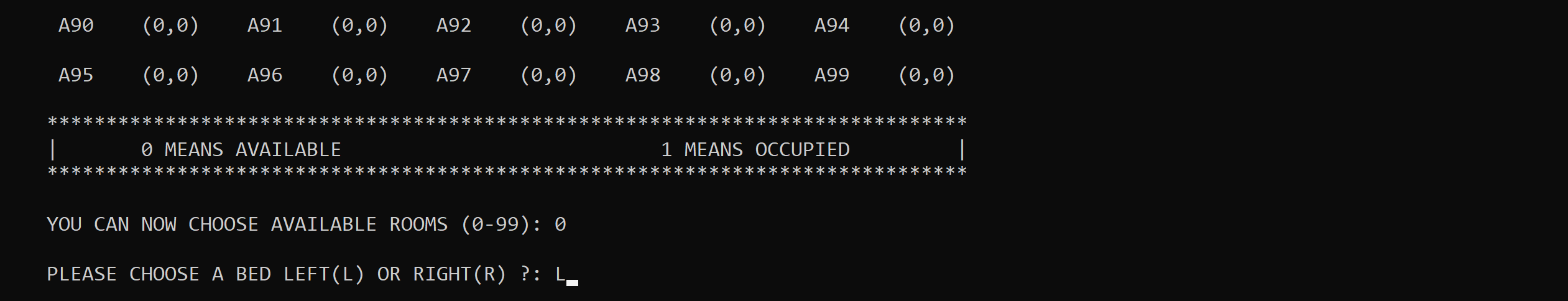
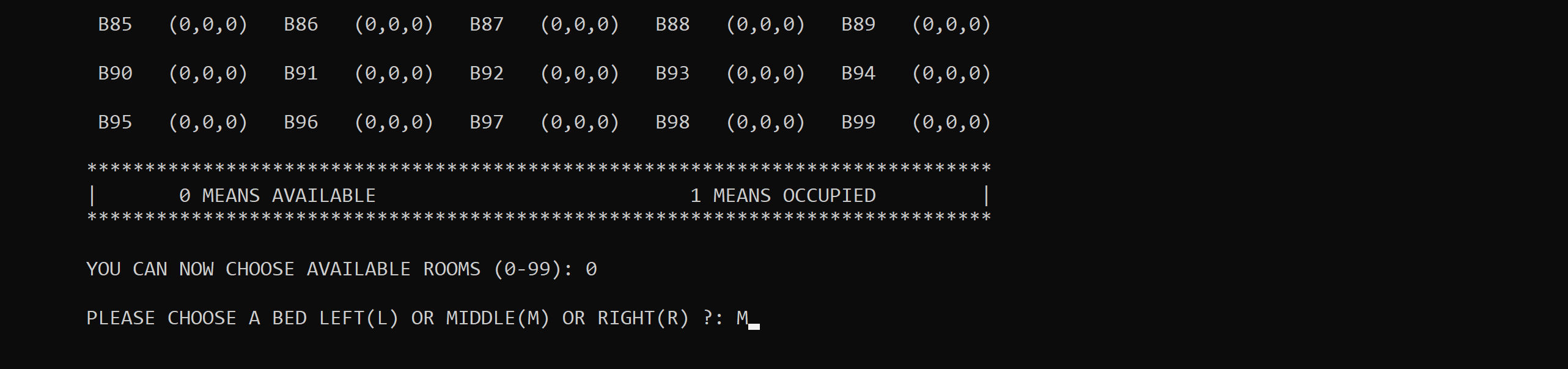
6.5.3 | Choosing Beds in a room

Figure 6.5.3.2 Choosing bed in Block B3

Figure 6.5.3.1 Choosing Bed in Block A2

 Choosing bed after choosing room is for both Block A2 and B3, as block A2 is a double bedroom and B3 is 3-bedded room, after choosing the room, the program will then request the user to input either L, M or R, which represents Left, Middle and Right. In block A2 is L and R, in block B3 is L, M and R. After user chooses their bed, program will then redirect users to the laundry subscription menu.

### 6.5.4 | Bed Chosen is Occupied

Figure 6.5.4.1 Bed Chosen is Occupied

If the bed that has been chosen by the user is occupied. The program will output “SORRY, THIS BED HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER BED”, and proceeds to request another input from the user.

## 6.6 | Service Subscription Menu

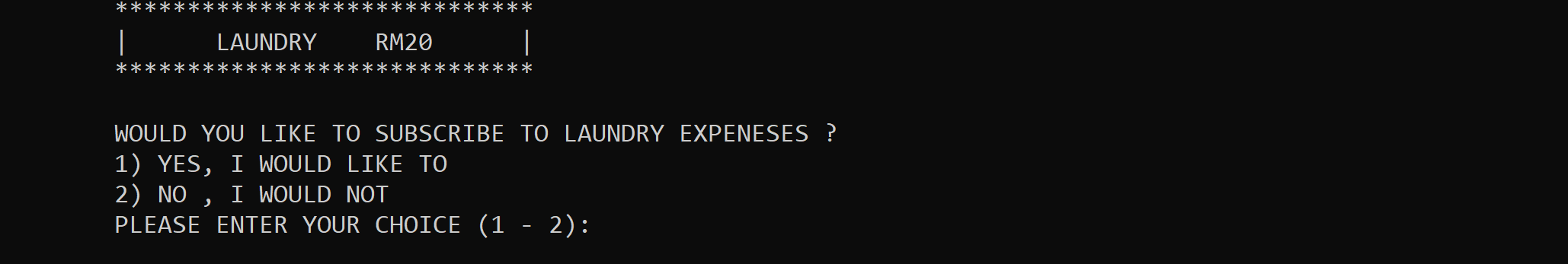
6.6.1 | Laundry Subscription Menu

Figure 6.6.1.1 Laundry Subscription

After being redirected from choosing the room number. The program will ask user if they want to add on laundry subscription and will request for an input of either 1. Yes or 2. No. After user input, the program will then show gym subscription menu.

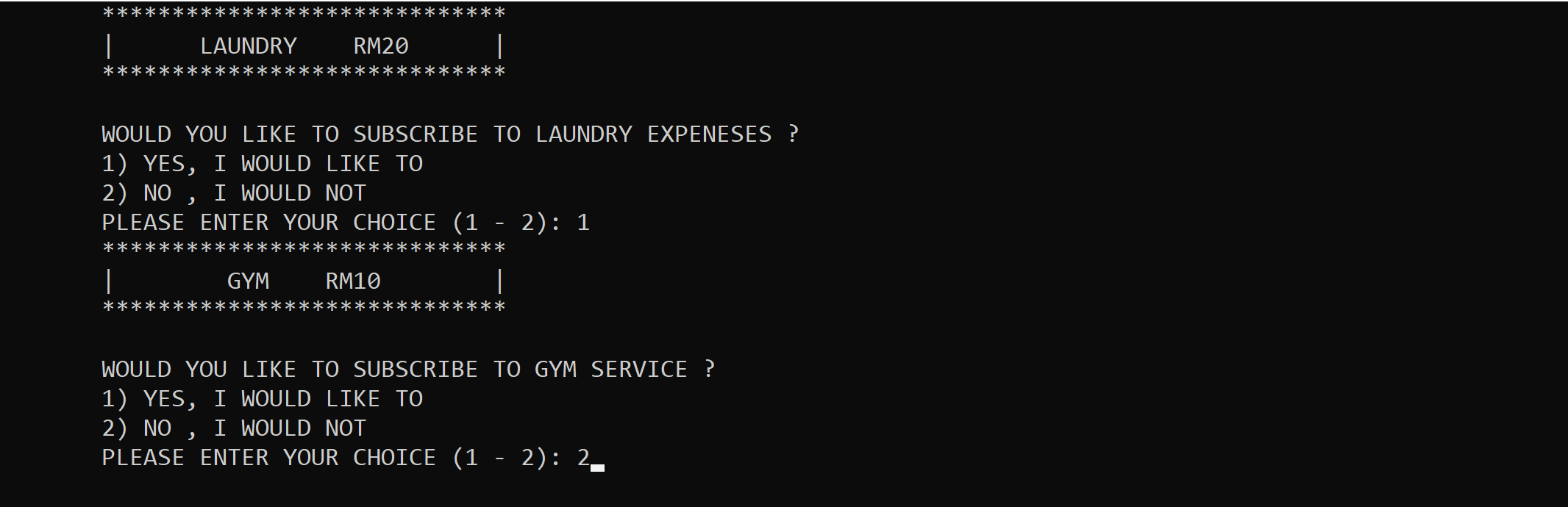
6.6.2 | Gym Subscription Menu

Figure 6.6.2.1: Gym Subscription

After being redirected from laundry subscription. The program will ask user if they want to add on gym subscription and will request for an input of either 1. Yes or 2. No.

6.6.3 | Week of Stay Given & Successful Booking

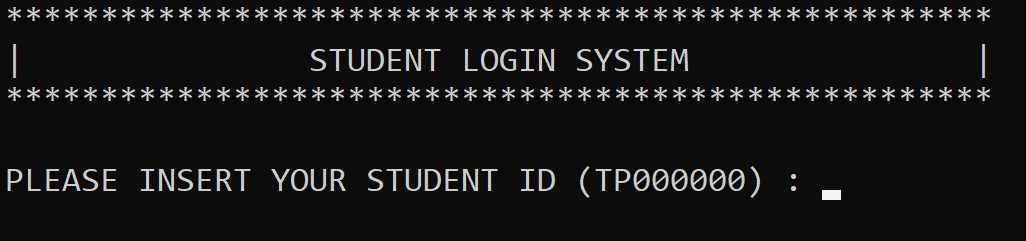
Figure 6.6.3.1 Given Week of Stay & Successful Booking

After gym subscription, the program will then ask the user to input the number of weeks they want to stay at the hostel. As shown above, after input the number of weeks user wanted to stay the program will then congratulate the user for successful booking and register thus output “PLEASE PROCEED TO LOGIN” to guide user to proceed to student login on the hostel management main menu and the program will redirect the user to the hostel management main menu.

## 6.7 | Student Login

Figure 6.7.2 If Enter Choice 2

Figure 6.7.1 If Enter Choice 2

 After the input 2, the program will then direct students to the student login system, for them to proceed login to check and change details as well as make payment. The program will then ask the user to input the student ID with the format TP000000.

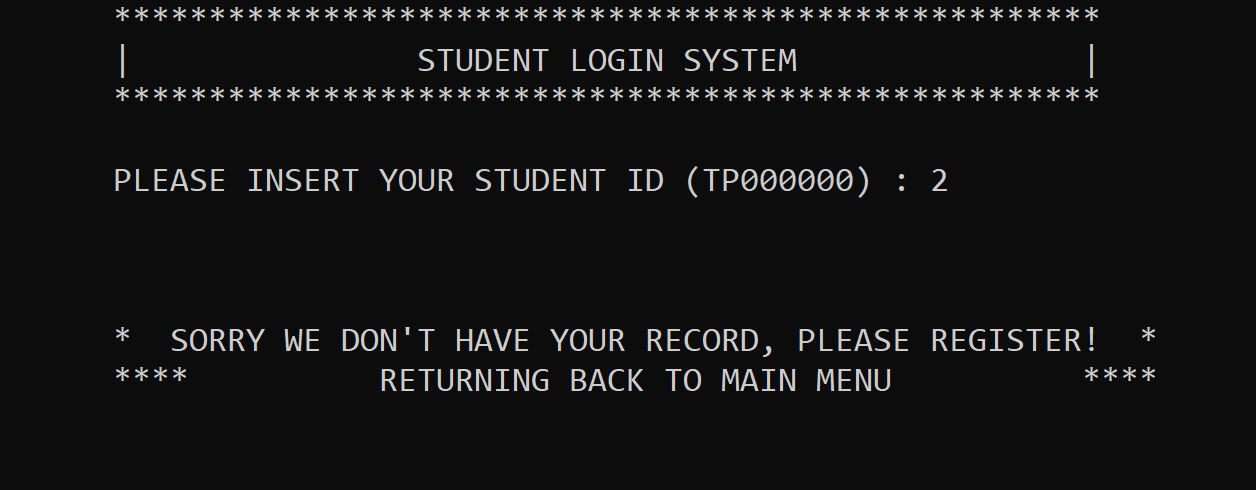
6.7.1 | No Record of Registration Found

Figure 6.7.1.1 If No Record of Registration Found

As shown in figure 6.7.1.1, If student inputs a student ID that is has not been registered, the system will then output a line of words, requesting unregistered user to register, and then the program will redirect back to hotel management system main menu.

### 6.7.2 | Record Found & Successful Login

Figure 6.7.2.1 Record Found & Successful Login

Figure 6.7.2.1 shows that if a record was found, the program will then say hi to the student and the program will then redirect the user to the login menu.

## 6.8 | Student Login Menu

Figure 6.8.1 Student Login Menu

As shown in figure 6.8.1, after being redirected from Student Login System, the program will then request another input from the user. Input 1 for change or cancel room, 2. Check and change personal details, 3. Make payment, 4 will be exit to the hostel management main menu.

## 6.9 | Change/ Cancel Room & Subscribe/ Unsubscribe Laundry or Gym Service

Figure 6.9.1 If Choice 1 Change / Cancel Room

If choice 1 is the given input, then the program will ask for another user input for either 1 change room 2 cancel room 3 change laundry or gym service as shown in figure 6.9.1.

### 6.9.1 | Change Room

Figure 6.9.1.2 After Insert 1 Redirect to Booking Menu

Figure 6.9.1.1 Choice 1 Change Room

After choosing choice 1 which is change room, the program will then redirect user to the booking menu like when they have successfully registered to change another room or change block. The menu showed in figure 6.9.1.2 is the male booking menu, because when student login into the system, the program read the information from text file and realize student is a male, so the program will redirect this student to male booking menu.

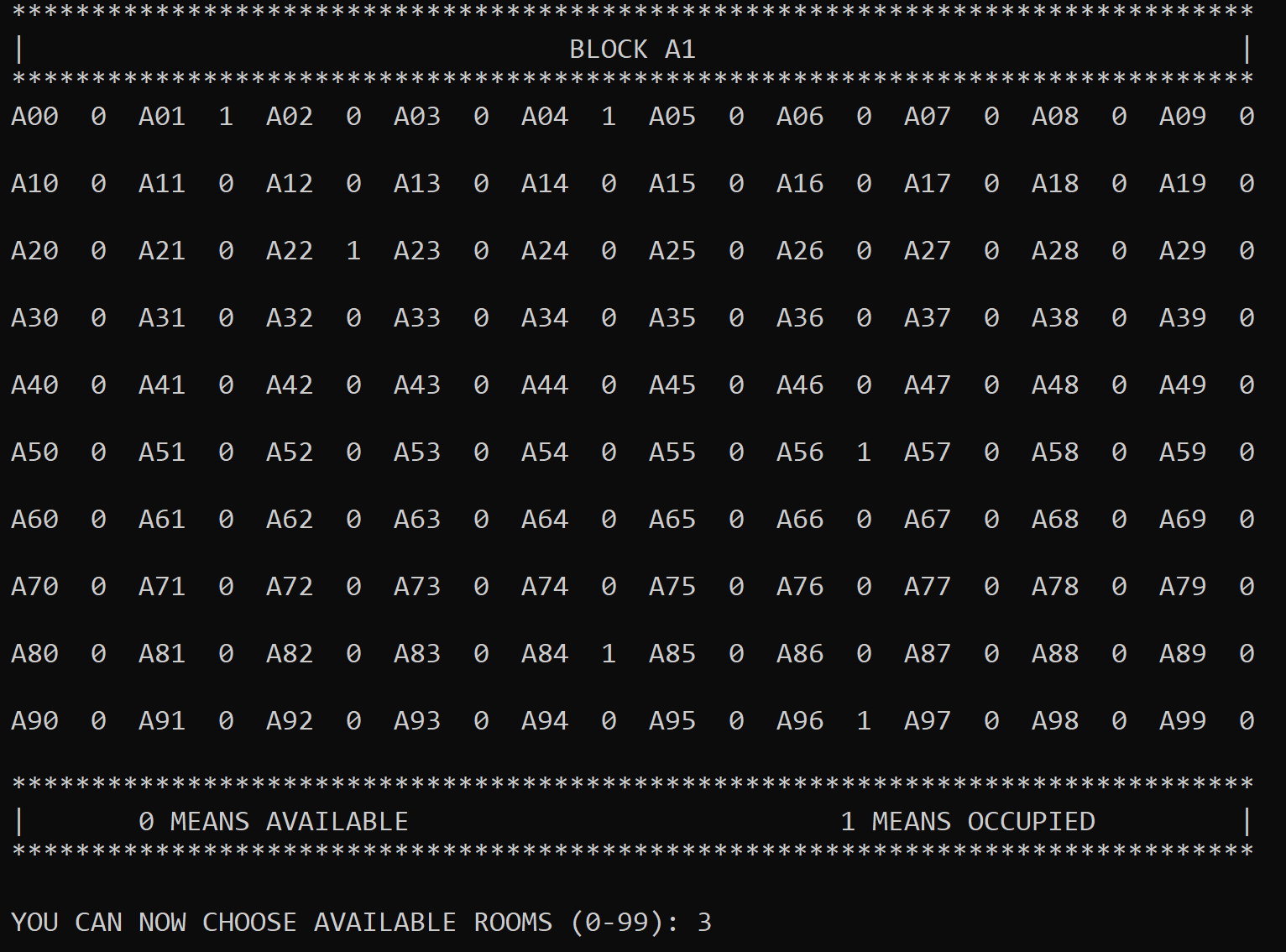
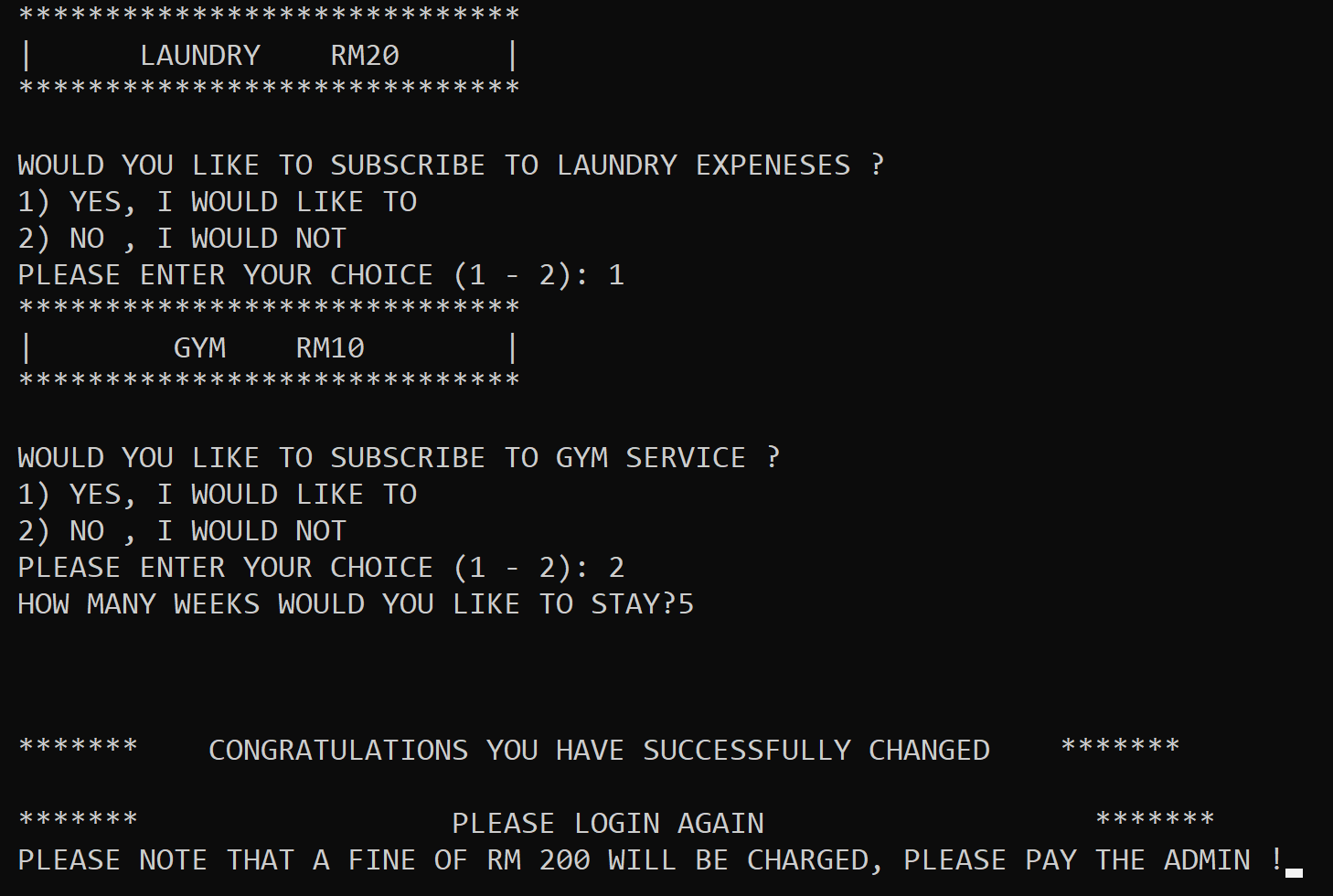
 As shown in figure 6.9.1.3 and figure 3.9.1.4, the process of booking a new room is the same as normal booking, but after everything is successfully, student will be asked to pay a fine of RM200 to the admin.

Figure 4.9.1.4 Booking New Room

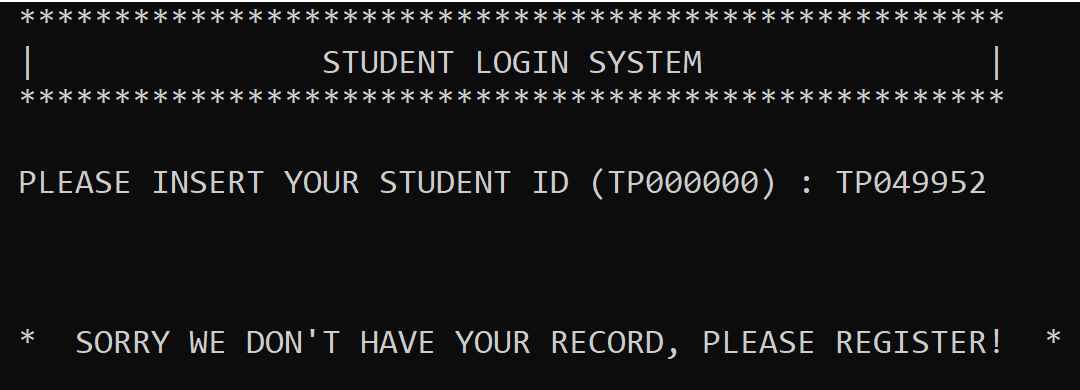
Figure 6.9.1.3 Choose Room to Change

### 6.9.2 | Cancel Room

Figure 6.9.2.3 Information Deleted after cancel room

Figure 6.9.2.2 Text displayed After Successfully Cancelled

Figure 6.9.2.1 Cancel Room

 After input choice 2 for cancel room as shown in figure 6.9.2.1, the program will then ask if the user really meant to cancel the room. If user inputs Y, the program will then print text that shows user has successfully cancelled the room as shown in figure 6.9.2.2. Afterwards, if student tries to login, there won’t be any record of the student again as shown in figure 6.9.2.3.

### 6.9.3 | Change Subscription Service

Figure 6.9.3.1 Change Laundry or Gym Service

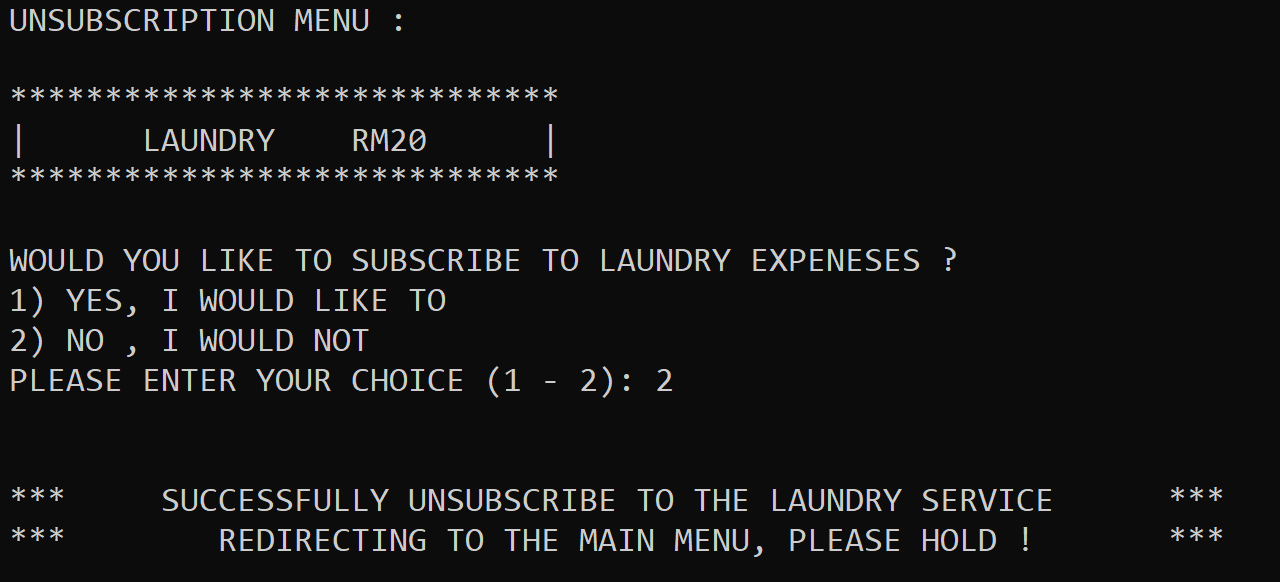
 If choice 3 is the input, the program will ask which subscription service the user wants to add as shown in figure 6.9.3.1. After the input, the program will then ask the user which service they like to add or cancel. Either laundry or gym.

Figure 6.9.3.2 Unsubscribe Laundry Service

If L was chosen, the program will then redirect users to the subscription or unsubscribe menu based on their previous choosing. As shown above in figure 6.9.3.2, the user has previously subscribed to laundry service, so he is redirected to unsubscribe menu. If they chose 2 not to subscribe the service again. The program will output that he had successfully unsubscribe to laundry service and then will be redirected to the main menu.

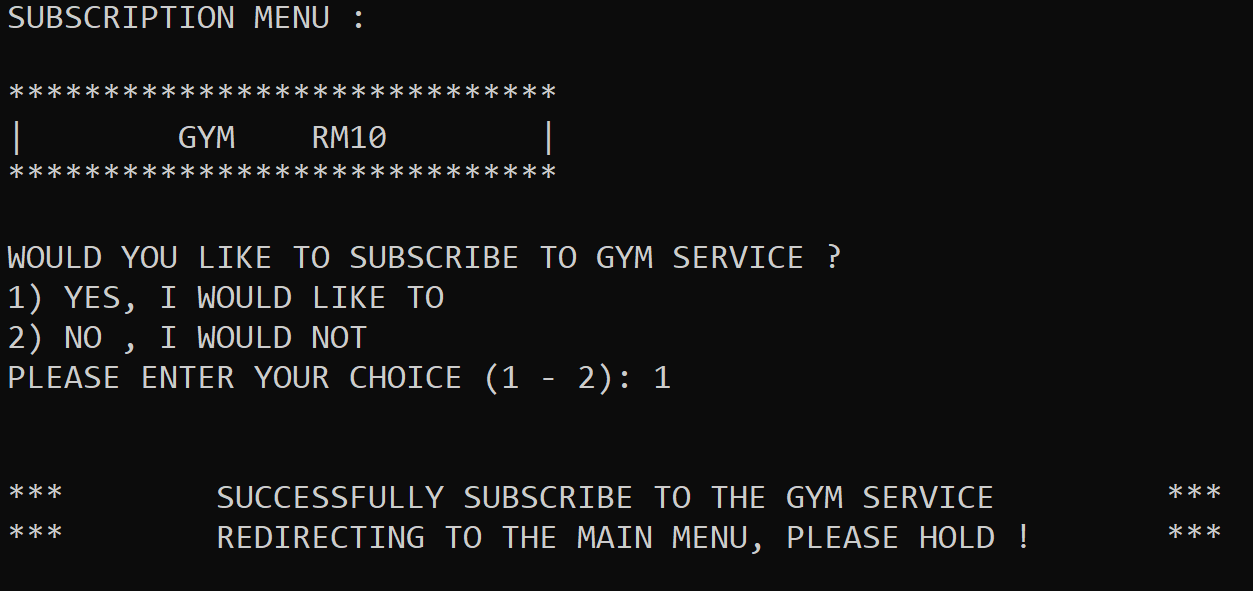
If G was chosen, the program will then redirect users to the subscription or unsubscribe menu based on their previous choosing. As shown above in figure 6.9.3.3, the user has previously not subscribed to gym service, so he is redirected to subscribe menu. If they chose 1 to subscribe the service. The program will output that he had successfully subscribe to gym service and then will be redirected to the main menu.

Figure 6.9.3.3 Subscribe to Gym Service

## 6.10 | Check / Change Personal Details

Figure 6.10.1 Choice 2 Check / Change Details

If user input is 2 as shown in figure 6.10.1, the program will then redirect the user to the personal detail menu.

## 6.11 | Personal Details Menu

Figure 6.11.1 Personal Details Menu

After being redirected from the student login menu, the personal details will show all the details of the student who login just now in the program. The information includes student ID, first name, last name, gender, identification and so on as shown in figure 6.11.1. The program will then request for an input asking if the user wants to change their personal details.

### 6.11.1 | Change Personal Details

Figure 6.11.1.1 Change Personal Details

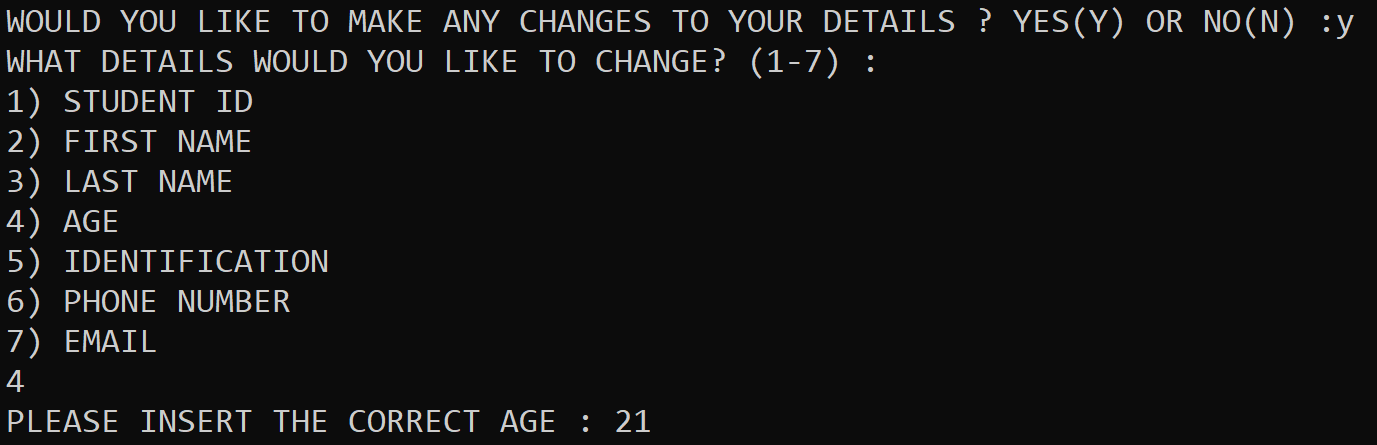
 If user inputs Y, the program will then ask user which information to change. As shown in figure 6.11.1.1, the user wants to change the age.

Figure 6.11.1.2 Changing Age

After choosing 4, the program will then request user to input the correct age as shown in figure 6.11.1.2. User input 21 as shown in figure 6.11.1.2.

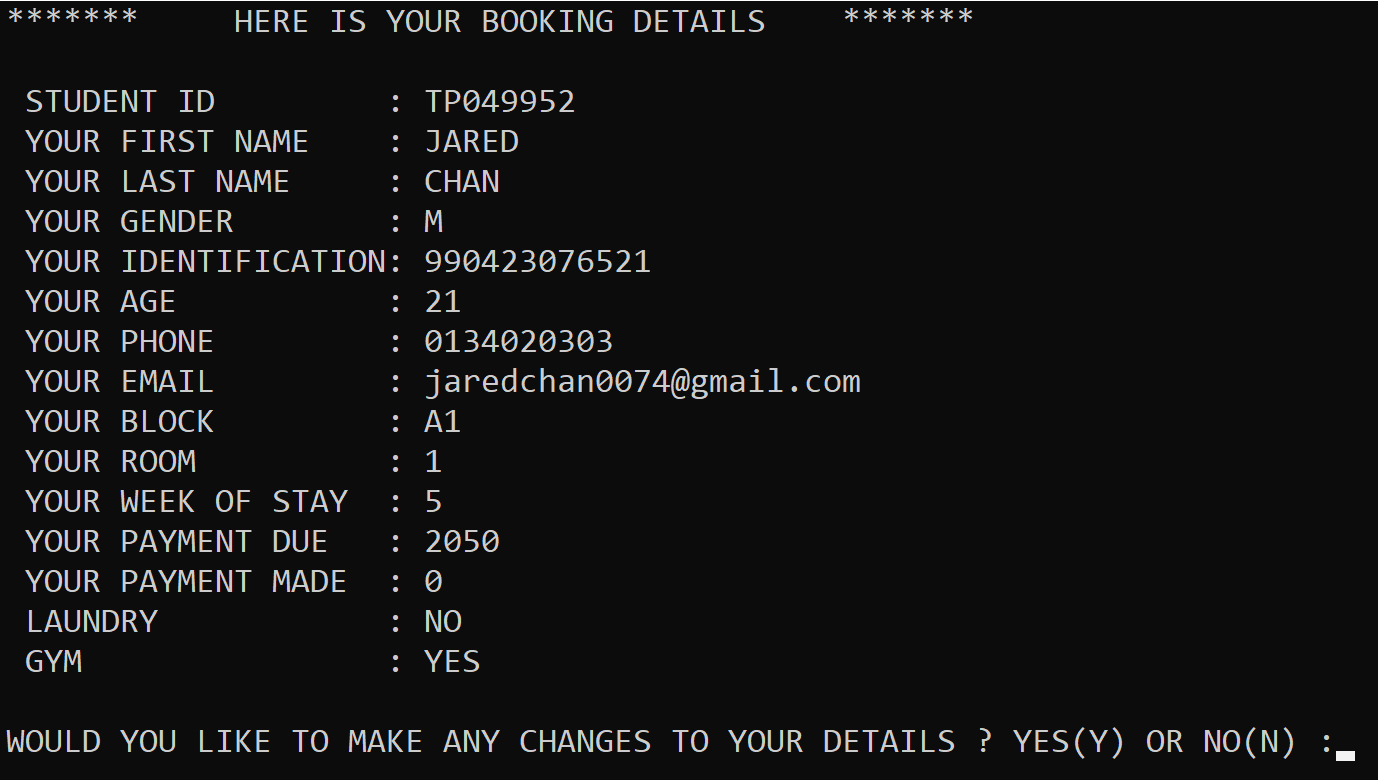
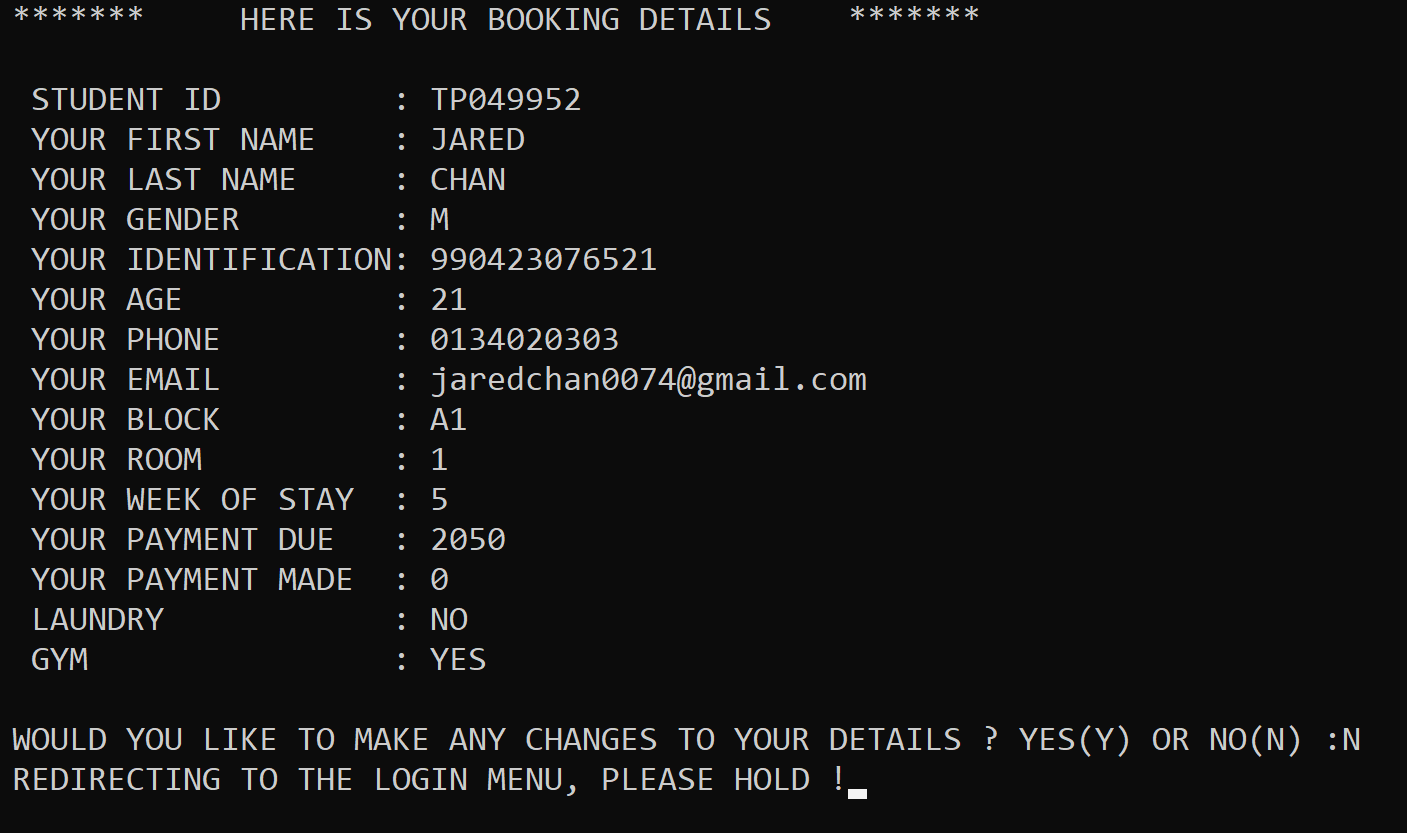
 After user input the correct age, the program will then change the age from 20 to 21 as shown in figure 6.11.1.3. The program can change the student ID, first name, last name, age, identification, phone number and email.

Figure 6.11.1.4 If User Chose Not to Change

Figure 6.11.1.3 Personal Detail Age Changed

If user inputs N in the personal detail menu, the program will redirect the user to the login menu as shown in figure 6.11.1.4.

## 6.12 | Payment Facility

Figure 6.12.2 If Choice 3

Figure 6.12.1 If Choice 3

If user inputs 3 when in the student login system, the program will then redirect user to the payment facility. The payment facility will show the amount due by the student and ask for them to make their payment.

### 6.12.1 | Payment Amount More Than Amount Due

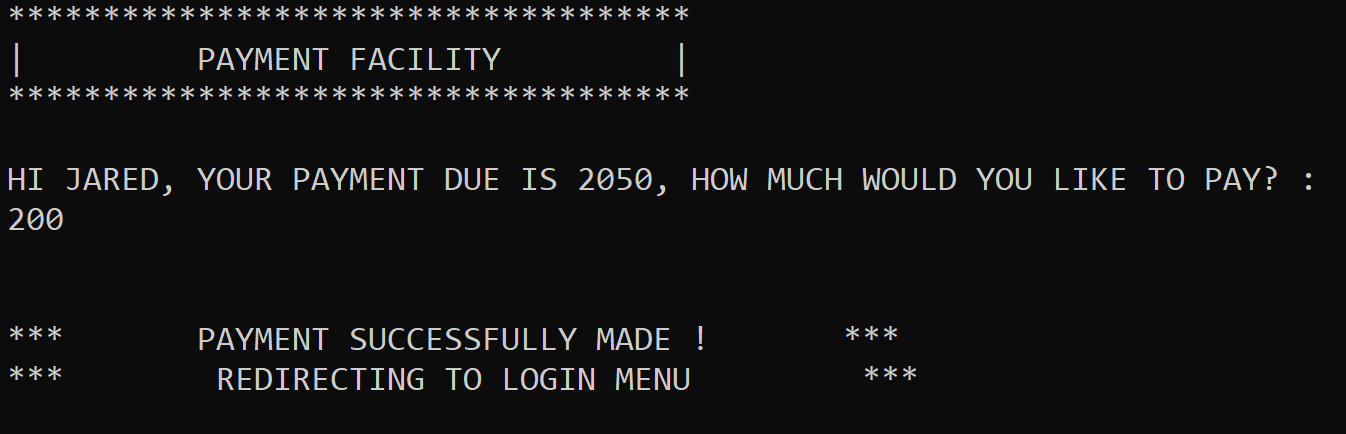
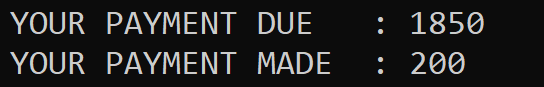
Figure 6.12.1.1 If Payment is more than amount due

If users chose to pay more than the amount due, the program will request the user to make a payment lesser than their amount due and will again request another amount from the user as shown in figure 6.12.1.1.

### 6.12.2 | Payment Successfully Made

Figure 6.12.2.2 Student Details changed after Payment successfully made

Figure 6.12.2.1 Payment Successfully Made

 After payment successfully made, the program will then update students’ details, and redirect the student to the login menu as shown in figure 6.12.2.1. After student detail has been updated the payment made will be added and the payment due will be updated based on payment made as shown in figure 6.12.2.2.

## 6.13 | Exit from Student Login System

Figure 6.13.1 Exit from Student Login System

If user inputs 4, the program will then redirect users back to the hotel management system main menu, as shown in the figure 6.13.1.

## 6.14 | Admin Login

Figure 6.14.2 Admin Login System

Figure 6.14.1 If Choice 3 Admin Login

If user chooses 3 as input, the program will then redirect the user to the admin login system as shown in figure 6.14.1 and figure 6.14.2.

### 6.14.1 | Admin Successfully Login

Figure 6.14.1.1 Successful Admin Login

As shown in figure 6.14.1.1, After successful login by entering password AbcD1234, the program will then redirect the user to the admin system.

## 6.15 | Admin System

Figure 6.15.1 Admin System

Figure 6.15.1 shows the admin system. Within admin system there are 4 choices which is 1 check overall details, 2. search for student details, 3. Check block room availability and 4. Exit to main menu.

### 6.15.1 | Check Overall Details

Figure 6.15.1.1 Check Overall Details

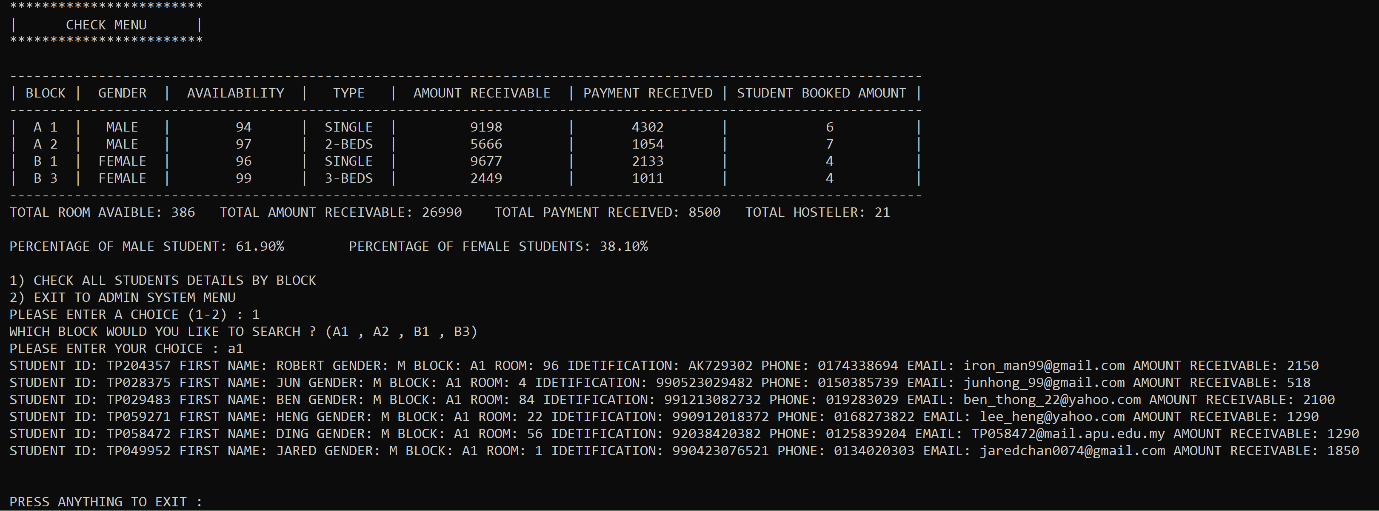
 After input 1 and redirect to the check overall menu, there will be a table stating all the necessary details, available room, amount receivable, payment received, student booked amount, total room available, total amount receivable, total payment received and total hosteler as well as the percentage of male and female students. The program will then request an input from user to either 1 list all the details of hosteler in the block or 2, exit to admin system menu.

Figure 6.15.1.2 Checking all hosteler details

After input one, the program will request the block that the admin wants to check. Figure 6.15.1.2 inputs a1, so the program will list all the details of the hosteler staying in block a1, including name, student ID as well as amount receivable and then ask for any input for user to exit to the admin system.

If user inputs 2 in the check menu, the program will redirect the user to the admin system as shown in figure 6.15.1.2.

Figure 6.15.1.2 Exit from Check Menu

### 6.15.2 | Search for Student Details

Figure 6.15.2.1 Student Search System

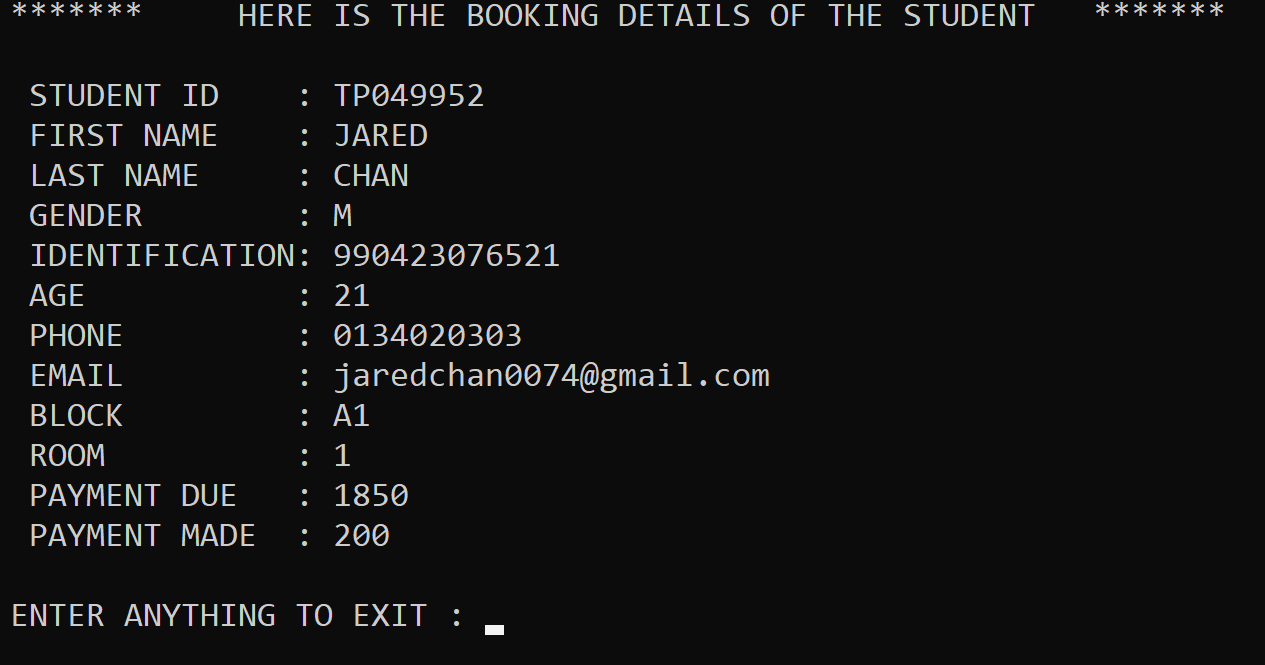
 After user inputs 2 in Admin System, the program will redirect admin to the student search system as shown in figure 6.15.2.1. The program will request for student ID to search.

Figure 6.15.2.2 Searched student details

After input student ID, if there is a record, the program will fetch the information from the text file and display all the relevant details as shown in figure 6.15.2.2 for the admin. After that, the program will request any input to exit to the admin system.

### 6.15.3 | Check Block Room Availability

Figure 6.15.3.2 Check Room Availability Visuals

Figure 6.15.3.1 Check Block Room Availability

After students input 3 for choice check block availability, the program will request an input from user to choose which block admin wants to view. Figure 6.15.3.1 shows that user input a1, then the program will display the whole block a1 to check available rooms as shown in figure 6.15.3.2. The program will then request any input to exit to the admin system.

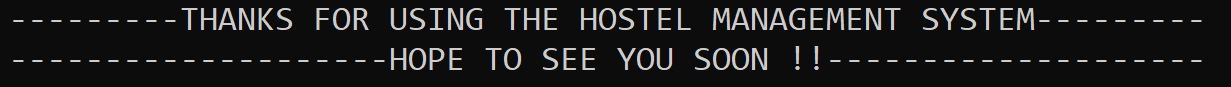
### 6.15.4 | Exit from Admin System

Figure 6.15.4.1 Exit from Admin System

After user input 4, the program will redirect user to hotel management system as shown in figure 6.15.4.1.

## 6.16 | Exit Program

Figure 6.16.1 Exit from Program

 After input 4 in the hostel management system, the program will exit the main menu and proceed to terminate the system.

# 7.0 | Conclusion

The Wisdom Hostel Management system was made by more than 20 functions each having its own unique features and functionality. The system contains 22 source file and one header file that contains all the function, thus 1 file for storing information of the students. The functionality of the system is ready to be utilized and it is ready to be presented to Wisdom College.

Process if developing this system has opened my eyes as if how system is develop based on simple yet hard steps like designing the program in pseudocode and flowchart, having to debug one of the most difficult logic errors. Overall, I’ve really learnt a lot from our Lecturer Mr. Sivaguru, as he had not only taught us how to program, but most of the time having detail explanations as if what skill should be needed and giving us a short glimpse of how the working environment works.

# 8.0 | References

Programiz, n.d. *C strcpy().* [Online]   
Available at: https://www.programiz.com/c-programming/library-function/string.h/strcpy  
[Accessed 10 June 2019].

Singh, C., 2015. *C – goto statement with example.* [Online]   
Available at: https://beginnersbook.com/2014/01/c-goto-statement/  
[Accessed 10 June 2019].

Tutorialspoint, n.d. *C library function - fscanf().* [Online]   
Available at: https://www.tutorialspoint.com/c\_standard\_library/c\_function\_fscanf.htm  
[Accessed 12 June 2019].

Tutorialspoint, n.d. *C library function - strcmp().* [Online]   
Available at: https://www.tutorialspoint.com/c\_standard\_library/c\_function\_strcmp.htm  
[Accessed 10 June 2019 ].