Programmming and software design

Submission

Submitted by: Abdelraoof Naser (ID: 197724)

Module Name: Computer Architecture

Module Code: 22COMP05I

Supervised by: Dr. sally hany

Computer Engineering Department, Degree Year 2 Egypt, April 2023

FIRST:Student Registration:

1)Registration Function:

A)Python coding and explanation:

```
def studentRegistration(occupants,A,B,masterRooms,replacedA,replacedB,replacedMaster):
    details = []
    name = input('Enter your name: ')
    TP = input('Enter your TP number: ')
    print("""Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room  """)
    details.append(name.upper())
    details.append(TP)
    i=1
    while i ==1:
        choice = input('Enter your choice(A or B): ')
        while choice.upper() != 'A' and choice.upper()!='B':
              print('Wrong input')
              choice = input('Enter your choice(A or B): ')
```

First,students are asked to enter their information (Name -TP).Then,rooms ,pricing, apartment types and other details are shown to the students to choose from them.

A while loop is created to enable the continue function which helps in case of typing a wrong input or the unavailability of a certain type of rooms.

To avoid error an early while loop is created to make sure students enter one of two option which are: A or B

This function depends on many lists which student details will be appended in them or lists that have checked-out students' names that are going to be replaced with the new occupants'

Every function carry one of the three room types name which are(A-B-master rooms)

```
    if room.upper() == 'A':
        rent = 500
        if len(masterRooms) <= 20:
            roomChoice=roomForMaster(masterRooms,room)
        else:
            if len(replacedMaster)>0:
                roomChoice=replacement(replacedMaster,masterRooms)
            else:
                print('Sorry, no available rooms in this type. Try a different one')
                continue
        i = 2
        details.append(str(rent))
        details.append(roomChoice)
        file = masterRooms
        choice = 'master'
        continue
    elif room.upper() == 'B':
        rent = 300
        if len(B) <= 40:
            roomChoice = roomNumber(B, choice)
        else:
            if len(replacedB) > 0:
                roomChoice = replacement(replacedB, B)
            else:
                print('Sorry, no available rooms in this type. Try a different one')
                continue
        i = 2
        details.append(str(rent))
        details.append(roomChoice)
        file = B
        continue
elif choice.upper() == 'A':
    rent = 400
    if len(A) <= 40:
        roomChoice = roomNumber(A, choice)
    else:
        if len(replacedA) > 0:
            roomChoice = replacement(replacedA, A)
        else:
            print('Sorry, no available rooms in this type. Try a different one')
            continue
    details.append(str(rent))
    details.append(roomChoice)
    file = A
    i = 2
    continue
```

Nested if statements are used to see which type is chosen and then assign details and call secondary functions which vary depending on the type.Every type of room has a certain number of units available .That's why the length of lists that carry unit occapants are checked. If there is no available , the program sees if there are any checked out students that can be replaced.if the checked out list contains information ,a secondery function is called to append the new name and assign a room number to the new occupant.If there are no rooms left,an apology is printed and the program let him check if other types have available rooms by using the continue function.

All the details and the return of the secondary function get assigned to a list.

```
            continue
details.append(choice.upper())
internet=internetSubscription(rent)
payment = paymentCalculation(rent,internet)
details.append(str(payment))
occupants.append(details)
file.append(details)
return details
```

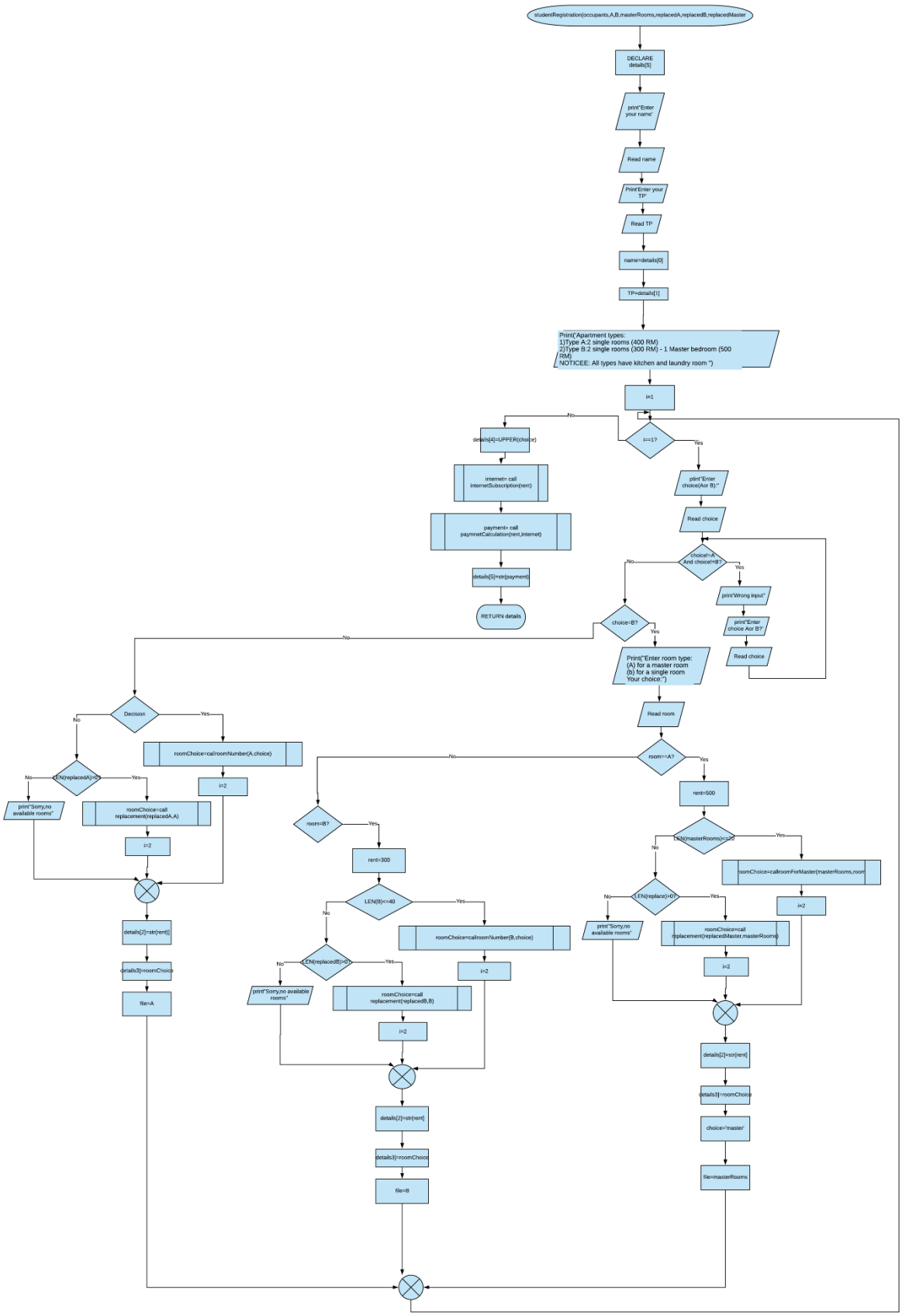
After assigning all the details to a list, this list gets appended in one the three type lists.

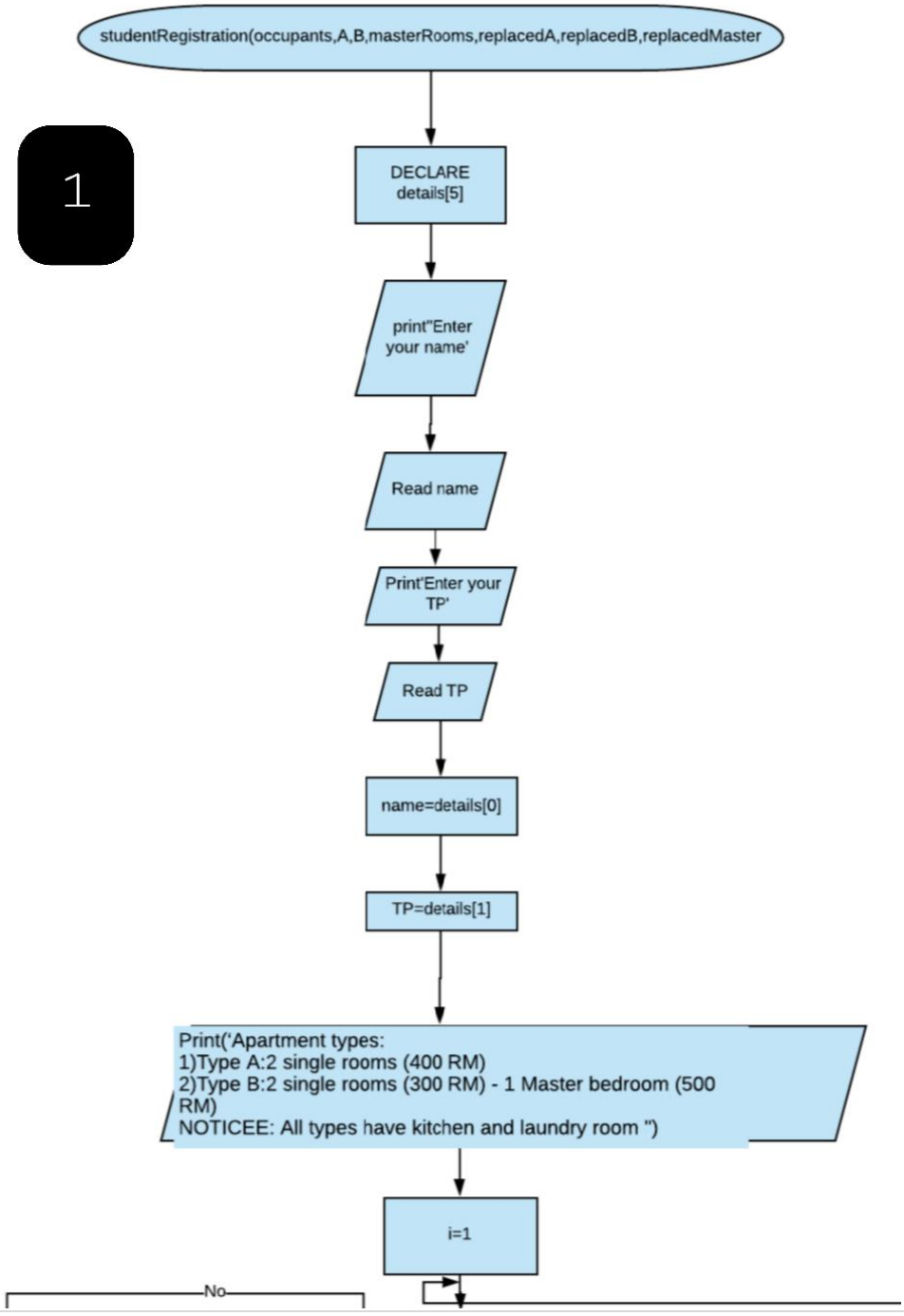
B) pseudo code:

```
    FUNCTION
studentRegistration(occupants,A,BmasterRooms,replacedA,replacedB,replacedMaster)
                    DECLARE details[5]
                    Print(‘Enter your name:’)
                    Read name
                    Name= details[0]
                    Print(‘Enter your TP: ‘)
                    Read TP
                    details[1]=TP
                    Print(‘Apartment types:
    1)  Type A:2 single rooms (400 RM)
    2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
    NOTICEE: All types have kitchen and laundry room ")
                   i=1
                   Dowhile i ==1:
                      Print('Enter your choice(A or B): ')
                      Read choice
                       DOWHILE choice!= 'A' AND choice!='B':
                            print('Wrong input')
                            Print('Enter your choice(A or B): ')
                            Read choice
                      IF choice == 'B' THEN
                          Print("Enter room type:
                (A) for a master room
                (b) for a single room
                Your choice:")
                          Read room
                          IF room == 'A’ THEN
                             rent = 500
                             IF LEN(masterRooms) <= 20 THEN
                                roomChoice=call roomForMaster(masterRooms,room)
                                i=2
                             ELSE:
                                IF LEN(replacedMaster) >0 THEN
                                     roomChoice= call
replacement(replacedMaster,masterRooms)
                                      i=2
                                ELSE:
                                     Print’’Sorry no available rooms in this type.Try a different
one’’
                                ENDIF
                             ENDIF
                             details[2]=str(rent)
                             details[3]=roomChoice
                             file=masterRoom
                             choice=’master’
                          ELSE:
                               IF room==’B’ THEN
                                  Rent=300
                                  IF LEN(B) <= 40 THEN
```

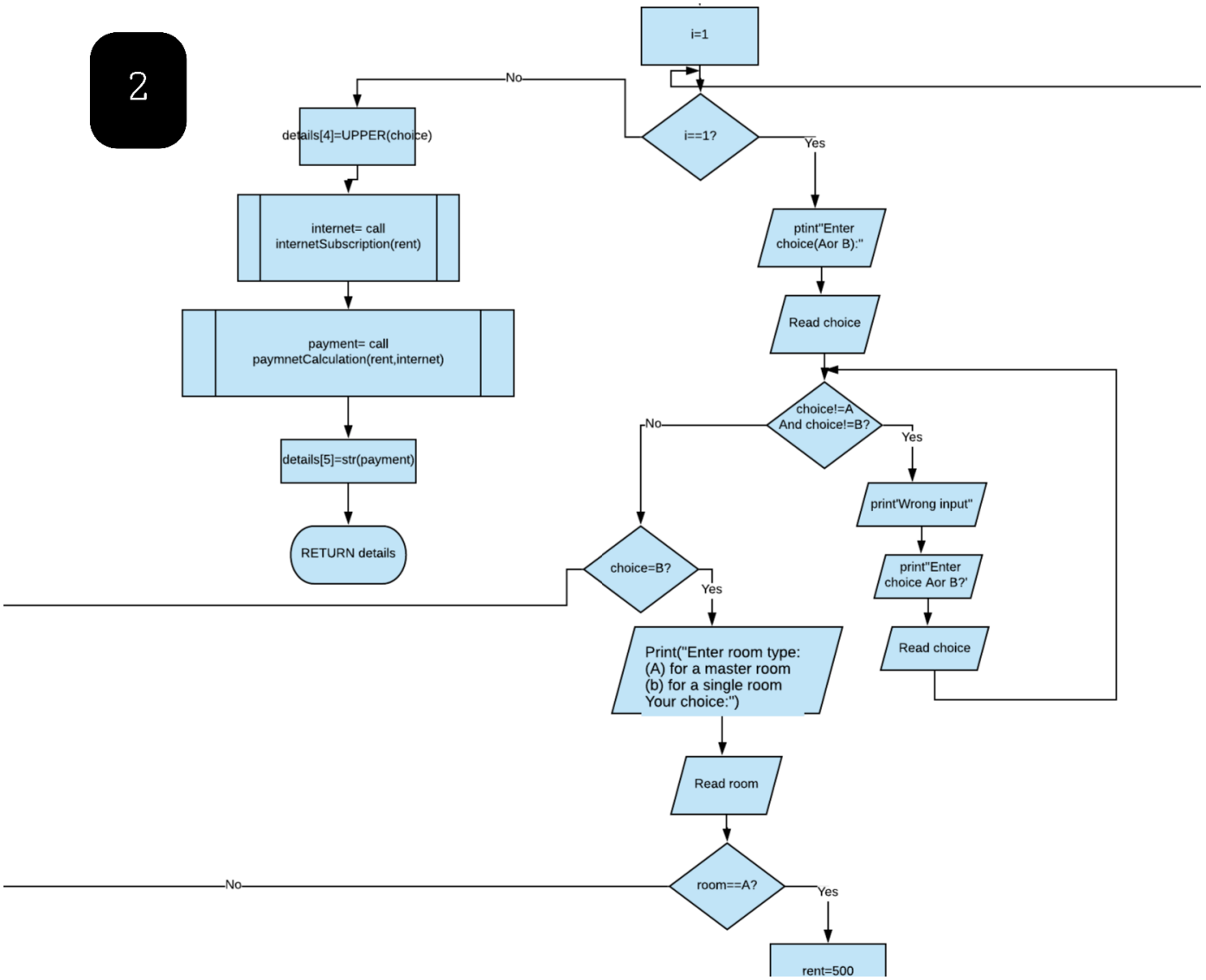
```
                        roomChoice= call roomNumber(B,choice)
                         i=2
                    ELSE:
                        IF LEN(replacedB)>0 THEN
                            roomChoice=call replacement(replaced,B)
                            i=2
                        ELSE:
                            Print''Sorry, no available rooms "
                        ENDIF
                    ENDIF
                    details[2]=str(rent)
                    details[3]=roomChoice
                    file=B
                ENDIF
            ELSE:
                rent=400
                IF LEN(A) <= 40 THEN
                    roomChoice= roomNumber(A,choice)
                    i=2
                ELSE:
                    IF LEN(replacedA) > 0 THEN
                        roomChoice=replacement(replacedA, A)
                        i=2
                    ELSE:
                        Print''Sorry, no available rooms in this type)
                    ENDIF
                ENDIF
                details[2]=str(rent)
                details[3]=roomChoice
                file=A
            ENDIF
        ENDDO
        Details[4]=UPPER(choice)
        Internet= call internetSubscription(rent)
        Payment= call paymentCalculation(rent,internet)
        Details[5]=str(payment)
        RETURN details
ENDFUNCTION
```

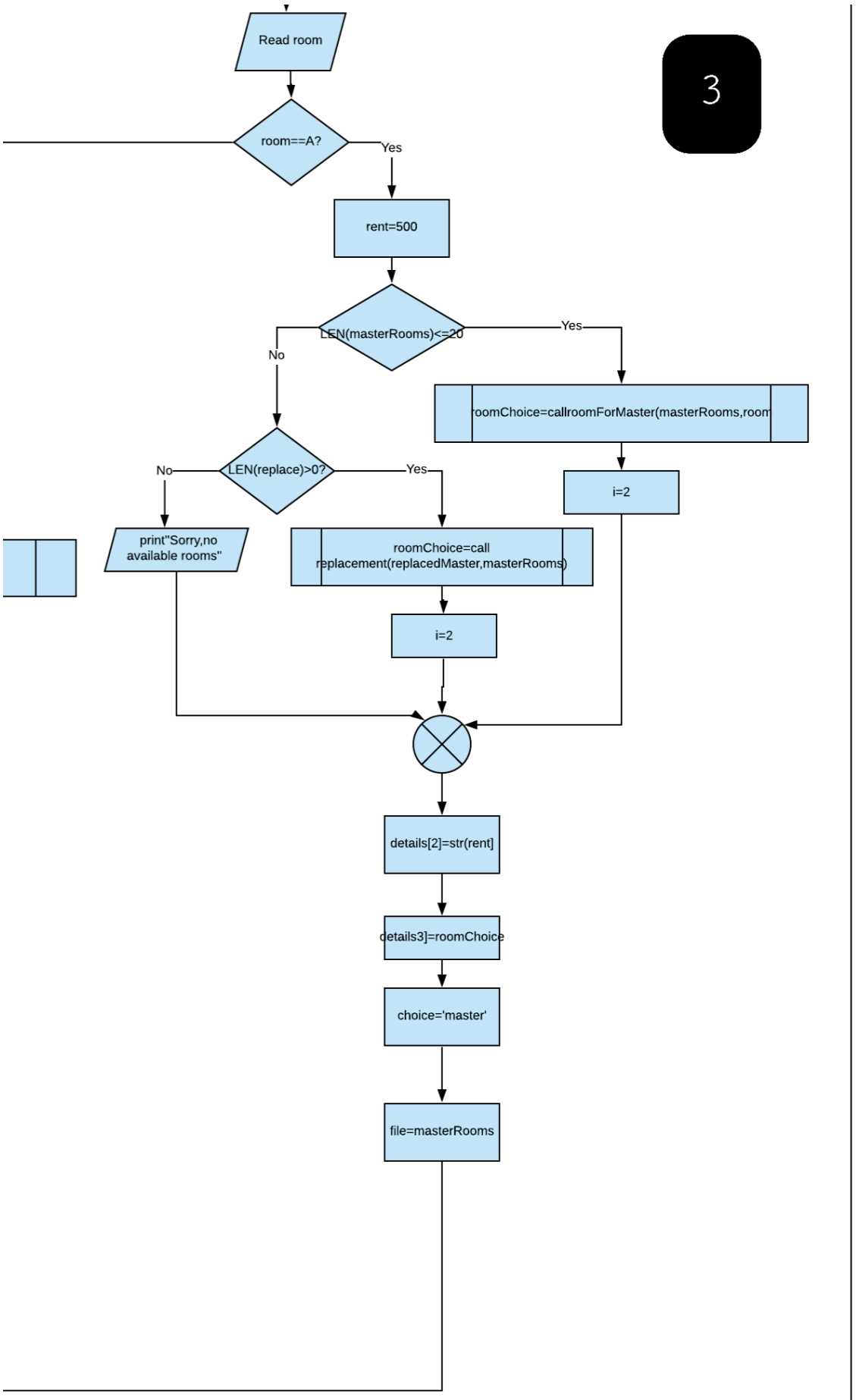
c)Flow chart:


studentRegistration(occupants,A,B,masterRooms,replacedA,replacedB,replacedMaster)
DECLARE details[5]
print"Enter your name"
Read name
Print"Enter your TP"
Read TP
name=details[0]
TP=details[1]
Print('Apartment types:
1)Type A:2 single rooms (400 RM)
2)Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room ")
i=1
i==1?
No
Yes
details[4]=UPPER(choice)
internet= call internetSubscription(rent)
payment= call paymnetCalculation(rent,internet)
details[5]=str(payment)
RETURN details
print"Enter choice(Aor B):"
Read choice
choice!=A And choice!=B?
print"Wrong input"
print"Enter choice Aor B?"
Read choice
choice=B?
Print("Enter room type:
(A) for a master room
(B) for a single room
Your choice:")
Read room
room==A?
rent=500
LEN(masterRooms)<=2
roomChoice=callroomForMaster(masterRooms,room
LEN(replace)>0?
print"Sorry,no available rooms"
roomChoice=call replacement(replacedMaster,masterRooms)
i=2
i=2
details[2]=str(rent)
details[3]=roomChoice
choice='master'
file=masterRooms
room=B?
rent=300
LEN(B)<=40
roomChoice=callroomNumber(B,choice)
i=2
LEN(replacedB)>0?
print"Sorry,no available rooms"
roomChoice=call replacement(replacedB,B)
i=2
details[2]=str(rent)
details[3]=roomChoice
file=B
Decision
roomChoice=callroomNumber(A,choice)
i=2
LEN(replacedA)>0?
print"Sorry,no available rooms"
roomChoice=call replacement(replacedA,A)
i=2
details[2]=str(rent)
details[3]=roomChoice
file=A

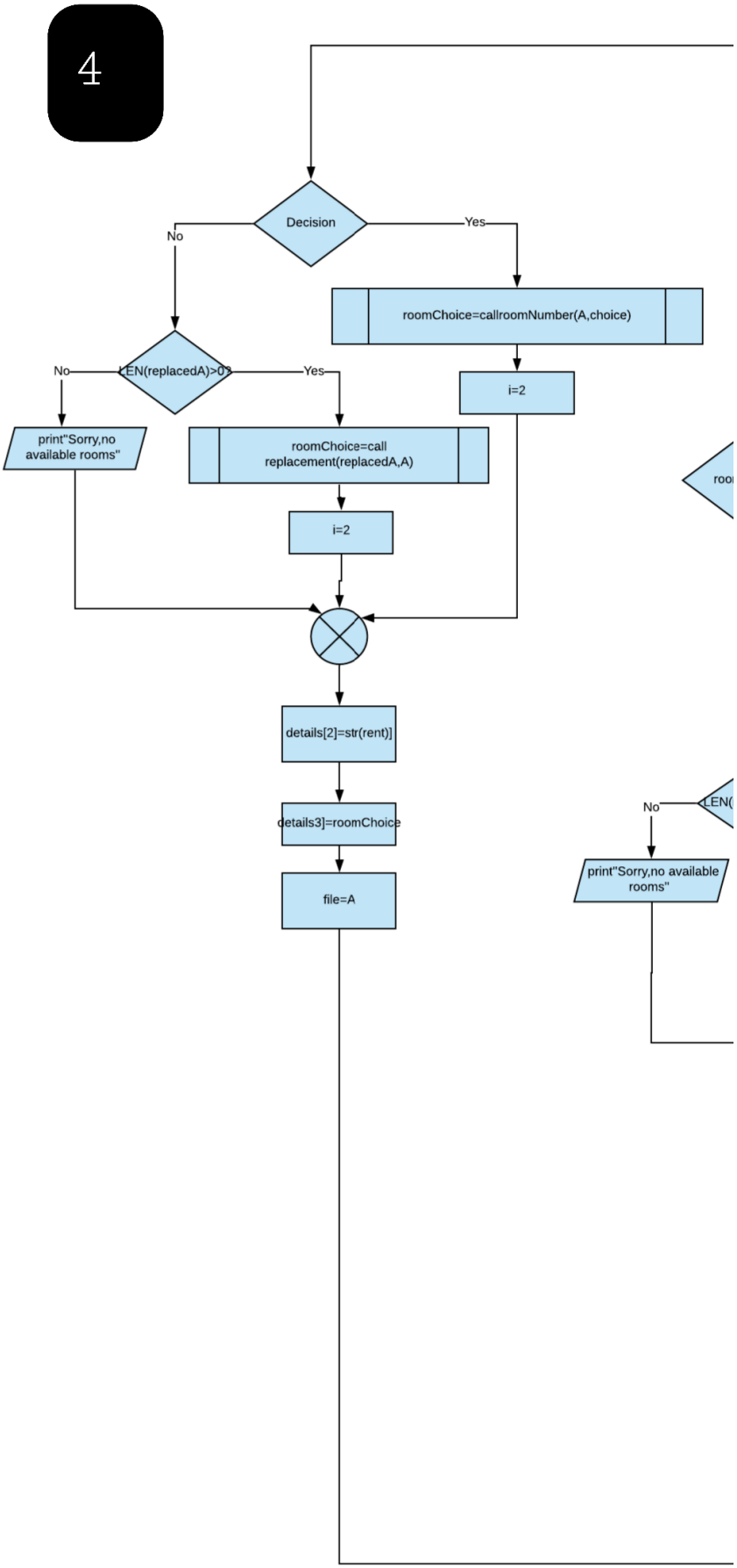

studentRegistration(occupants,A,B,masterRooms,replacedA,replacedB,replacedMaster
1
DECLARE details[5]
print"Enter your name'
Read name
Print'Enter your TP'
Read TP
name=details[0]
TP=details[1]
Print('Apartment types:
1)Type A:2 single rooms (400 RM)
2)Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room ")
i=1
No

2


i=1
i==1?
No
Yes
details[4]=UPPER(choice)
internet= call internetSubscription(rent)
payment= call paymnetCalculation(rent,internet)
details[5]=str(payment)
RETURN details
ptint"Enter choice(Aor B):"
Read choice
choice!=A And choice!=B?
No
Yes
print'Wrong input"
print"Enter choice Aor B?'
Read choice
choice=B?
Yes
Print("Enter room type:
(A) for a master room
(b) for a single room
Your choice:")
Read room
room==A?
No
Yes
rent=500


3
Read room
room==A?
Yes
rent=500
LEN(masterRooms)<=20
Yes
No
roomChoice=callroomForMaster(masterRooms,room
LEN(replace)>0?
No
Yes
i=2
print"Sorry,no available rooms"
roomChoice=call replacement(replacedMaster,masterRooms)
i=2
details[2]=str(rent]
details3]=roomChoice
choice='master'
file=masterRooms

4


Decision
No
Yes
roomChoice=callroomNumber(A,choice)
No
LEN(replacedA)>0?
Yes
i=2
print"Sorry,no available rooms"
roomChoice=call replacement(replacedA,A)
i=2
details[2]=str(rent)]
details3]=roomChoice
file=A
No
print"Sorry,no available rooms"

2)Internet subscription Function:

A) Python coding and explanation:

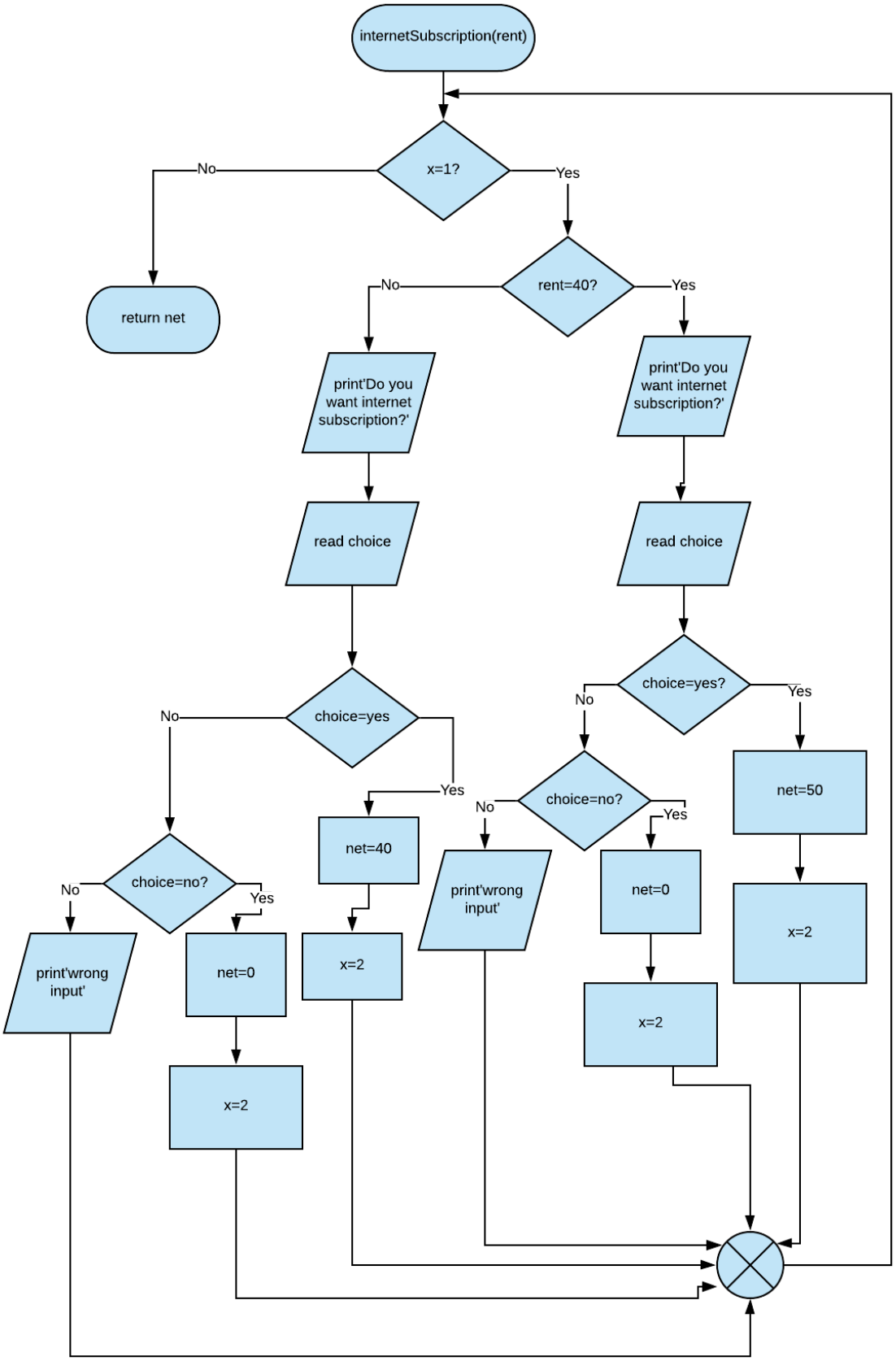
```
def internetSubscription(rent):
    x = 1
    while x == 1:
        if rent == 400:
            internet = input('Do you want internet connection (50 RM) [yes or no]: ')
            if internet.lower() == 'yes':
                net = 50
                x = 2
                continue
            elif internet.lower() == 'no':
                net = 0
                x = 2
                continue
            else:
                print('Invalid input')
                continue
        else:
            internet = input('Do you want internet connection (40 RM) [yes or no]: ')
            if internet.lower() == 'yes':
                net = 40
                x = 2
                continue
            elif internet.lower() == 'no':
                net = 0
                x = 2
                continue
            else:
                print('Invalid input')
                continue
    return net
```

If statements are used to assign the internet price depending on the room type.
Lower() function is used to accept the choice entered either it's in lower or upper casing.
The value of x is changed if the choice is accurate and continue() function is used to end the while loop without checking other if statement which makes the coding more functional.

B) pseudo code:

```
FUNCION internetSubscription(rent):
        X=1
        DOWHILE X==1
          IF rent==400 THEN:
            Print'' Do you want internet subscription? ‘’
            Read choice
            IF choice==’yes’ THEN:
              net=50
              x=2
             ELSE:
                 IF choice==’’no’’ THEN:
                    net=0
                    x=2
                 ELSE:
                      Print ’’Invalid input’’
                 ENDIF
           ELSE:
                Print ’’Do you want internet subscription? :’’
                Read choice
                IF choice==’yes’ THEN:
                   net=40
                   x=2
                ELSE:
                      IF choice==’no’ THEN:
                         net=0
                         x=2
                       ELSE:
                         Print’’Invalid input’’
                      ENDIF
                ENDIF
        ENDDO
        RETURN net
END FUNCTION
```

c)Flow chart:


internetSubscription(rent)
x=1?
No
Yes
return net
rent=40?
No
Yes
print'Do you want internet subscription?'
print'Do you want internet subscription?'
read choice
read choice
choice=yes
choice=yes?
No
Yes
No
Yes
net=40
choice=no?
No
Yes
print'wrong input'
net=0
net=50
choice=no?
No
Yes
print'wrong input'
net=0
x=2
x=2
x=2
x=2

3)Room numbering Function:

A) Python coding and explanation:

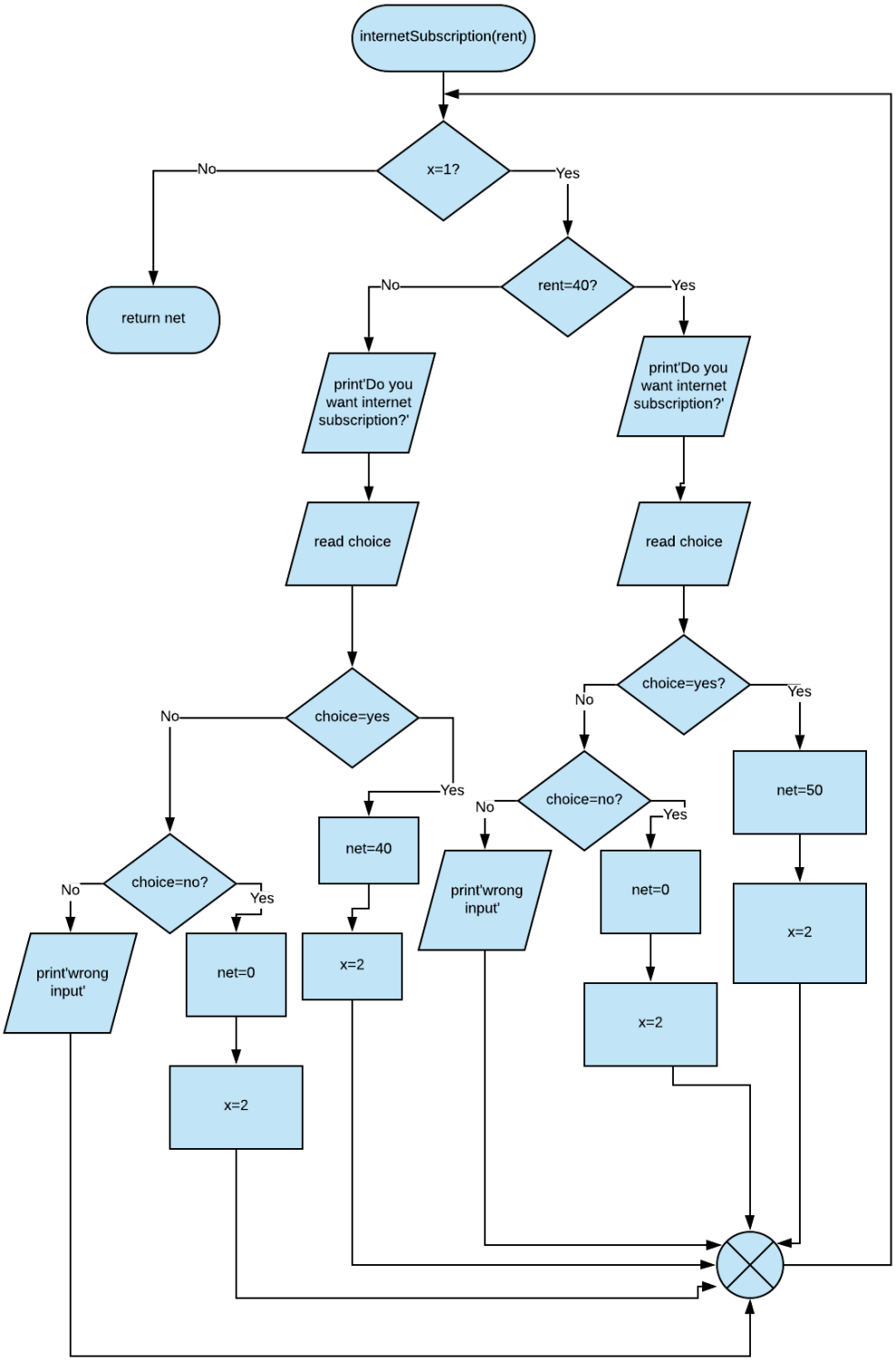
```
def roomNumber(list,type):
    apType=type
    calc=[]
    for i in range(1,21):
     calc.append(i)
     calc.append(i)
    numb=int(calc[len(list)])
    if len(list)%2==0:
        roomNum=f'{apType}-{numb}-1'
    else:
        roomNum=f'{apType}-{numb}-2'
    return roomNum
```

For loop is used to append values from (1 to 20) which are the apartment numbers .the value of the variable (I) is appended twice to the list because every apartment in either type A or B has 2 single rooms.The list that contains occupants information in the chosen type is put in the place of (list) to see how many occupants are there in this type.The value of room number is calculated by checking the length of the chosen list and then assigning the value in the calc list(that has a position number of the list's length) as avariable called (numb).the number of the occupants is divided by two.If there is a remainder that means room number two is vacant in an apartment.If there is no remainder that that means no vacant room in any of the occupied apartments.

B) pseudo code:

```
FUNCTION roomNumber(list,type):
        apType=str(type)
        DECLARE calc [40]
        X=0
        LOOP i FROM 1 TO 20 STEP 1:
                calc[x]=i
                calc[x+1]=i
                X = X+2
        ENDLOOP
        numb=int[calc(LEN(list)]
        IF len(list)%2==0 THEN
           roomNum=str(type)+'-'+str(numb)+'-1'
        ELSE
              roomNum=str(type)+'-'+str(numb)+'-2'
        RETURN roomNum
ENDFUNCTION
```

c) flow chart:


internetSubscription(rent)
x=1?
No
Yes
return net
rent=40?
No
Yes
print'Do you want internet subscription?'
print'Do you want internet subscription?'
read choice
read choice
choice=yes
choice=yes?
No
Yes
No
Yes
choice=no?
net=40
choice=no?
net=50
No
Yes
No
Yes
print'wrong input'
net=0
print'wrong input'
net=0
x=2
x=2
x=2
x=2

4)Room numbering for master rooms Function:

A) Python coding and explanation:

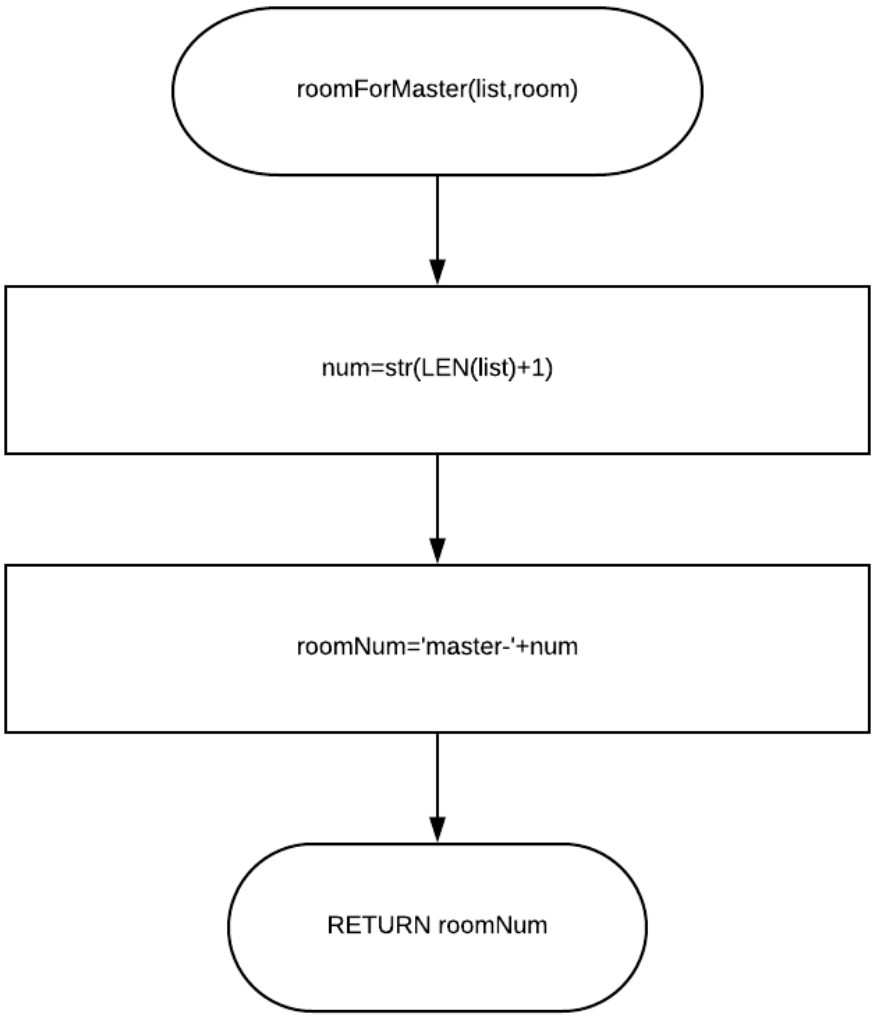
```
def roomForMaster(list,room):
    num=str(len(list)+1)
    roomNumb=f'master-{num}'
    return roomNumb
```

The number of occupants is calculated bt checking the length of the master room list.

B) pseudo code:

```
FUNCTION roomForMaster(list,room)
        Num=str(LEN(list)+1)
        roomNum=’master-’+num
       RETURN roomNum
ENDFUNCTION
```

c) Flow chart:



5) claulating the amount must be paid Function:

A) Python coding and explanation:

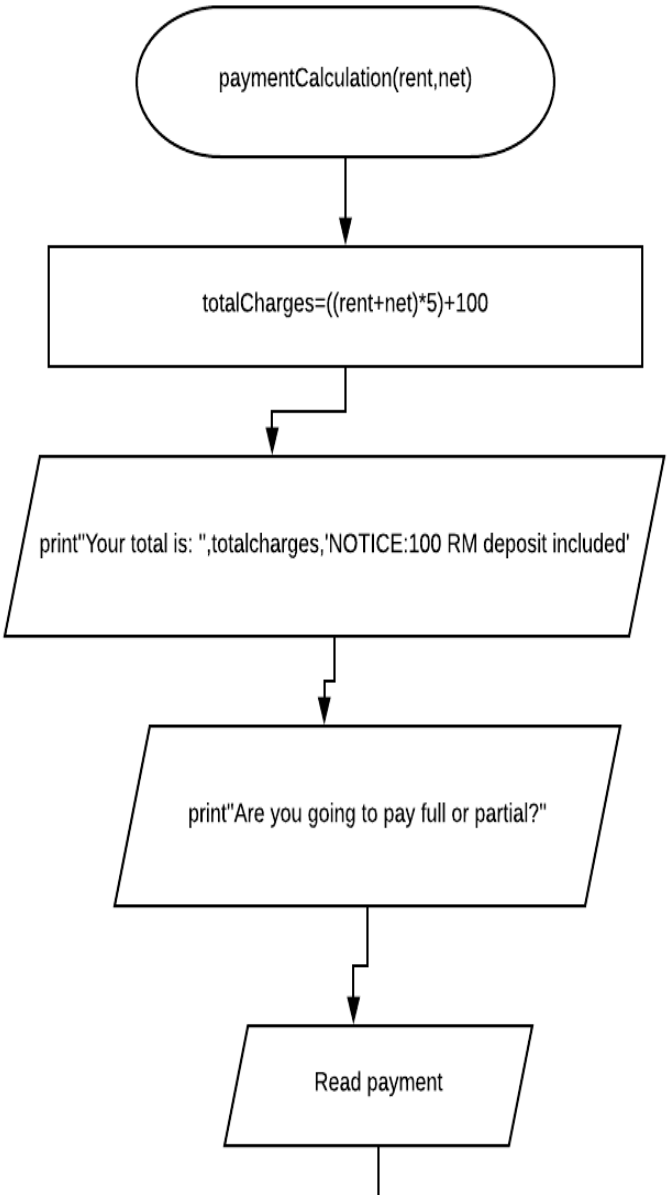
```
def paymentCalculation(rent,net):
    totalCharges = ((rent + net) * 5) + 100
    print('Your total is: ', totalCharges, '[Notice: 100 RM deposit included]')
    payment = input('Are you going to pay full or partial? :')
    while payment.lower() != 'partial' and payment.lower() != 'full':
          print("wrong input")
          payment = input('Are you going to pay full or partial? :')
    if payment.lower() == 'partial':
        print('You are required to pay at least 50% of total rent + 100 RM deposit')
        paid = int(input('Enter the amount you are going to pay: '))
        while not paid >= ((rent + net) * 2.5) + 100:
            print('Wrong input')
            paid = int(input('Enter the amount you are going to pay: '))
        totalPaid = paid
    if payment.lower() == 'full':
        totalPaid = totalCharges
    return totalPaid
```

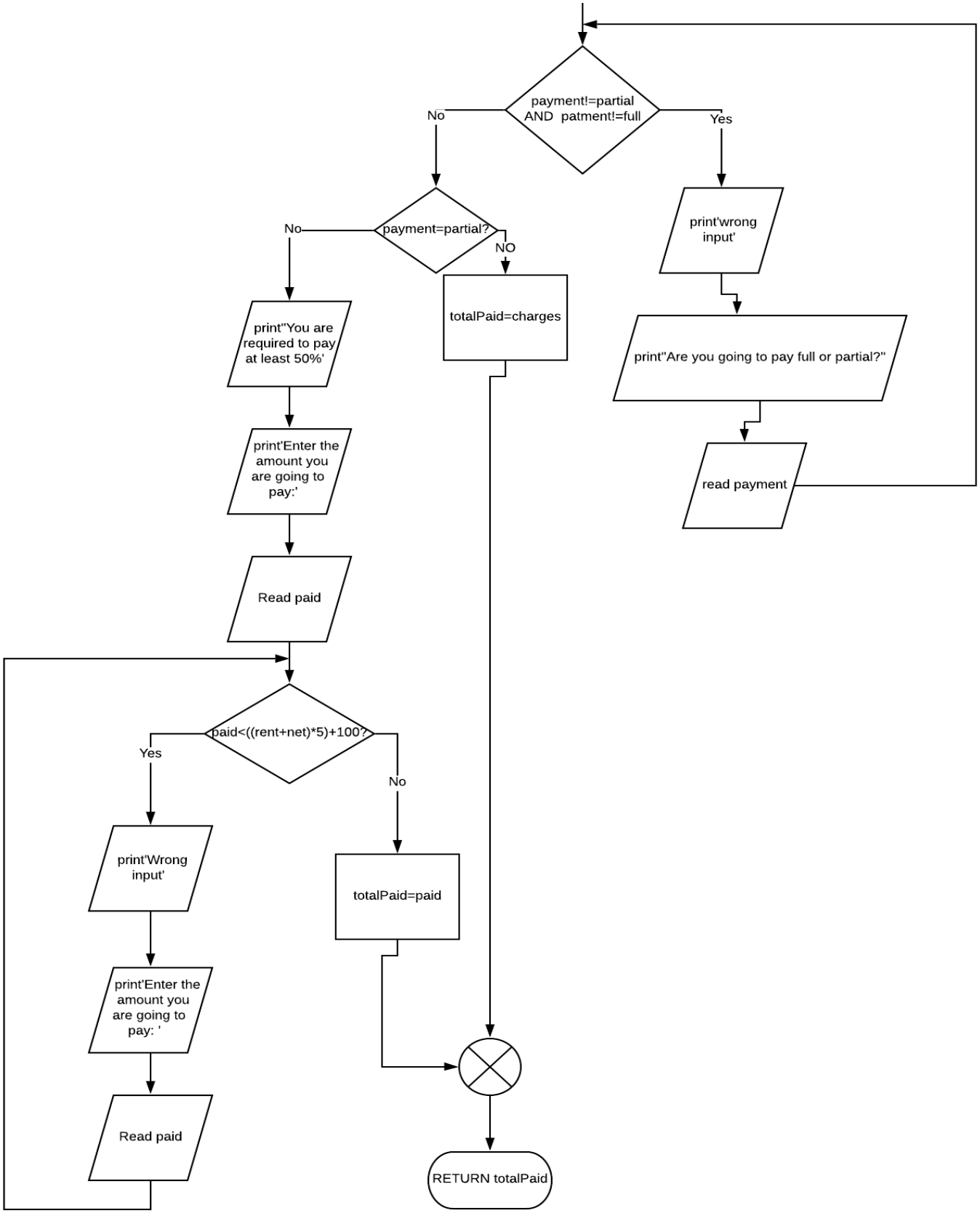
If the students is going to pay fully,the rent and the intenet subscription get added together and then get multiplied by five as every semester has five months.if the student is going to pay partial , he gets asked to enter the amount he is willing to pay. If the value that the student entered is less than 50% of the total amount ,the program doesn't accept the value and the student gets asked to enter ahigher value.

B) pseudo code:

```
FUNCTION paymentCalculation(rent,net):
        totalCharges=((rent+net)*5)+100
        Print''Your total is :'',totalCharges,''NOTICE:100 RM deposit included''
        Print''Are you going to pay full or partial ?''
        Read payment
        DOWHILE payment != 'partial' AND payment !='full':
              Print ''Wrong input''
              Print''Are you going to pay full or partial ?''
              Read payment
        ENDDO
        IF Payment =='partial' THEN
            Print''You are required to pay at least 50% of total rent + 100 RM deposit"
            Print'' Enter the amount you are going to pay: "
            Read paid
            DOWHILE paid < ((rent + net) * 2.5) + 100:
                  Print''Wrong input''
                  Print'' Enter the amount you are going to pay: "
                  Read paid
            ENDDO
            totalPaid=paid
        ELSE:
            TotaPaid=totalCharges
        ENDIF
        RETURN totalPaid
ENDFUNCTION
```

c) Flow chart:


paymentCalculation(rent,net)
totalCharges=((rent+net)*5)+100
print"Your total is: ",totalcharges,'NOTICE:100 RM deposit included'
print"Are you going to pay full or partial?"
Read payment


payment!=partial AND patment!=full
No
Yes
payment=partial?
No
NO
print'wrong input'
print"You are required to pay at least 50%'
totalPaid=charges
print"Are you going to pay full or partial?"
print'Enter the amount you are going to pay:'
read payment
Read paid
paid<((rent+net)*5)+100?
Yes
No
print'Wrong input'
totalPaid=paid
print'Enter the amount you are going to pay: '
Read paid
RETURN totalPaid

6) Saving information Function:

A) Python coding and explanation

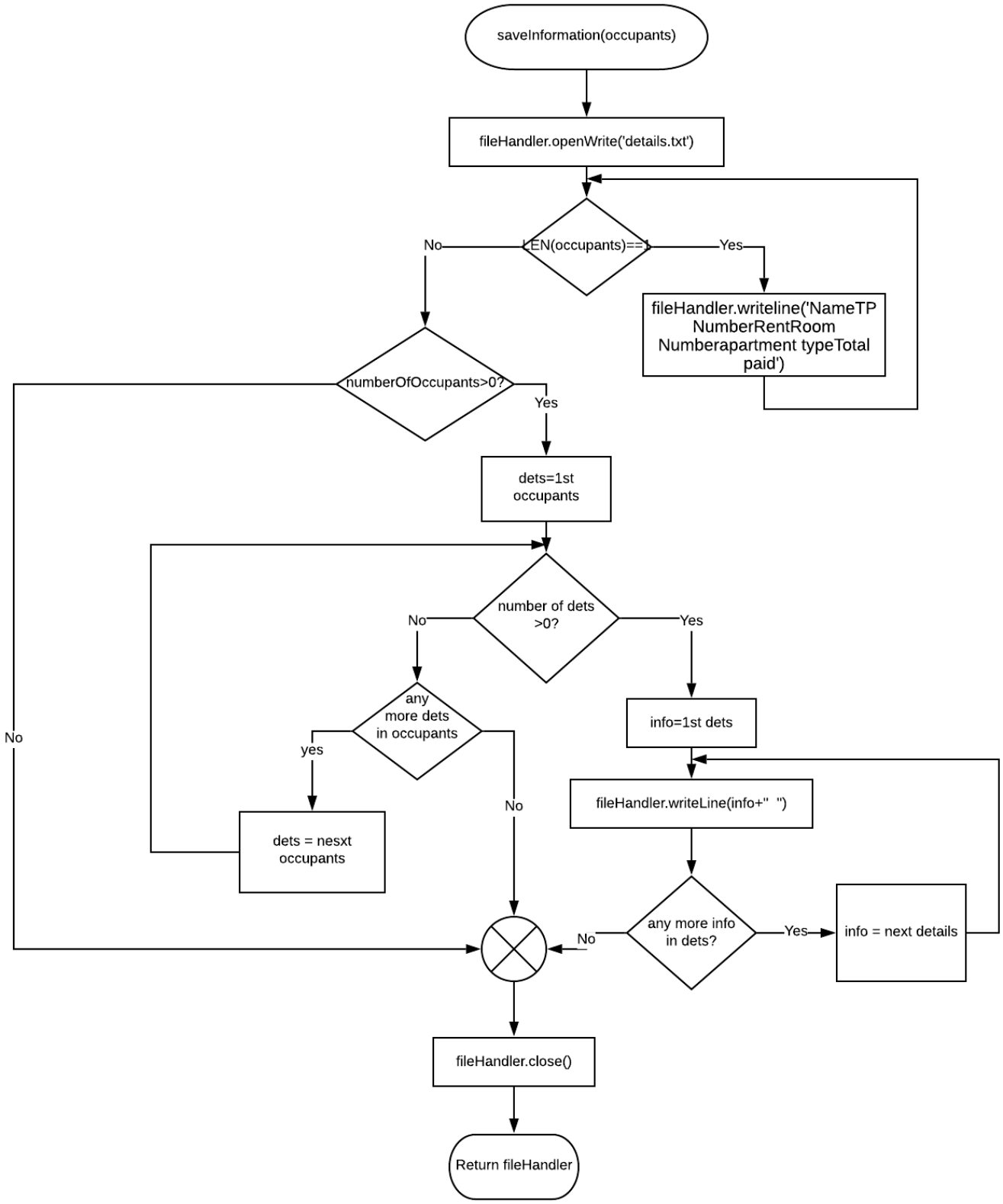
```
def saveInformation(occupants):
    fileHandler=open('details.txt','w')
    if len(occupants)==1:
        fileHandler.write('Name          TP Number        Rent       Room Number       apartment type    Total paid')
        fileHandler.write('\n')
    for det in occupants :
        for info in det:
            fileHandler.write(info)
            fileHandler.write('\t\t')
        fileHandler.write('\n')
    fileHandler.close()
    return fileHandler
```

After registering the students and filling their details, thier information gets appended in (occupants) list. that lists contains many secondary lists.Every secondary list contains the information of one student.this information gets saved in a text file called details.For loop is used to write the details in the secondary lists one by one .

B) pseudo code:

```
FUNCTION saveInformation(occupant):
        fileHandler=OpenWrite('details.txt')
        IF LEN(      occupants)==1 THEN
            fileHandler.writeline(  'Name          TP Number        Rent       Room Number
apartment type    Total paid')
        ENDIF
        FOR EACH det in occupants:
            FOR EACH info in det:
                fileHandler.writeline(info+''      '')
            ENDFOR
        ENDFOR
        fileHandler.close()
       RETURN fileHandler
ENDFUNCTION
```

c) Flow chart:


saveInformation(occupants)
fileHandler.openWrite('details.txt')
LEN(occupants)==1
No
Yes
fileHandler.writeline('NameTP NumberRentRoom Numberapartment typeTotal paid')
numberOfOccupants>0?
No
Yes
dets=1st occupants
number of dets >0?
No
Yes
any more dets in occupants
yes
No
dets = nesxt occupants
info=1st dets
fileHandler.writeLine(info+" ")
any more info in dets?
No
Yes
info = next details
fileHandler.close()
Return fileHandler

7) Replacement function Function:

A) Python coding and explanation

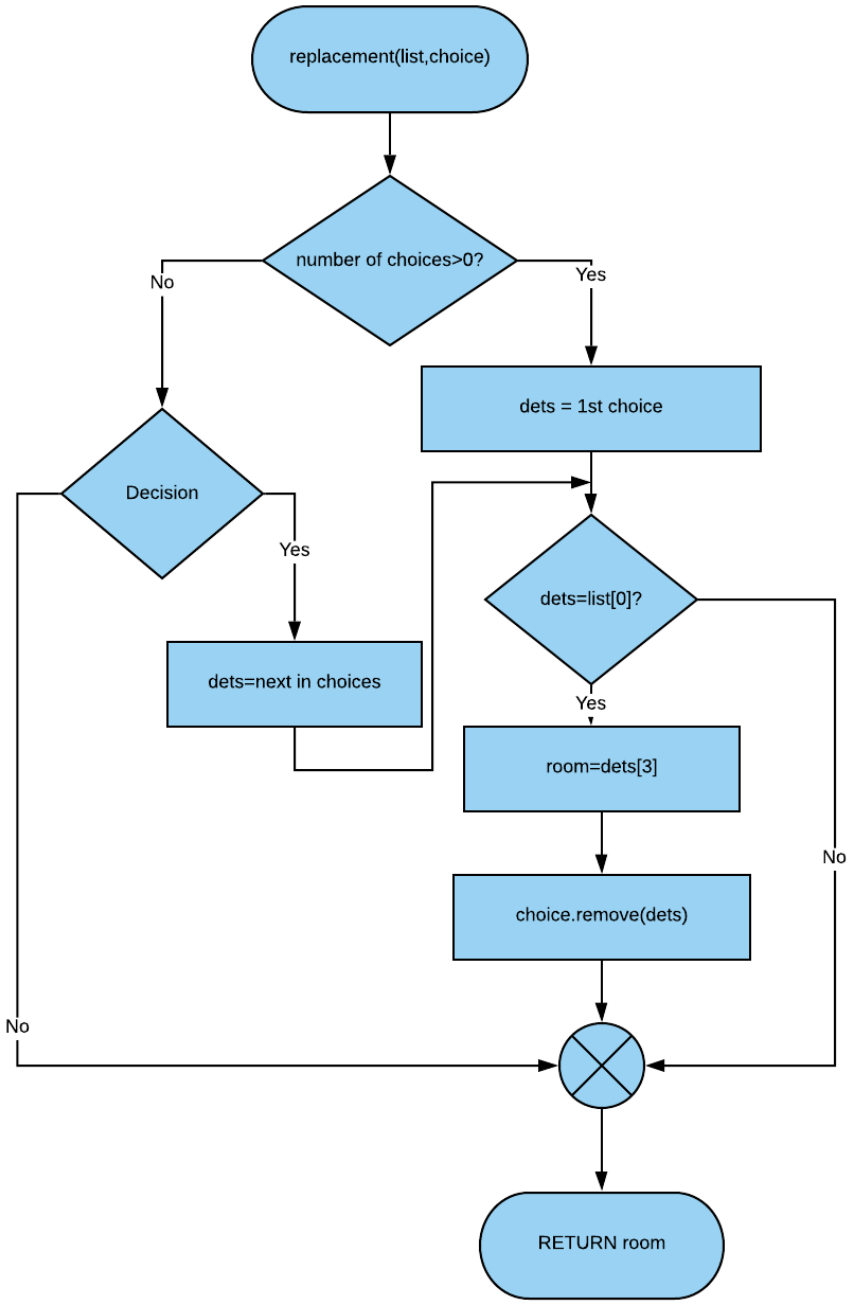
```
def replacement(list,choice):
    for dets in choice :
        if dets==list[0]:
            room=dets[3]
            choice.remove(dets)
    return room
```

This function is for finding new occupants for rooms whose occupants have checked out . For loop is used to match the checked out information with one of the secondary list so that his information can be romoved and replaced by someone else.

B) Pseudo code:

```
FUNCTION replacement(list,choice):
   FOR EACH dets in choice :
      iF dets==list[0] THEN
         room=dets[3]
         choice.remove(dets)
   RETURN room
ENDFUNCTION
```

B) Flow chart:


replacement(list,choice)
number of choices>0?
No
Yes
dets = 1st choice
Decision
Yes
dets=list[0]?
dets=next in choices
Yes
room=dets[3]
No
choice.remove(dets)
No
RETURN room

SECOND:Searching:

1) Searching by name :

A) Python coding and explanation:

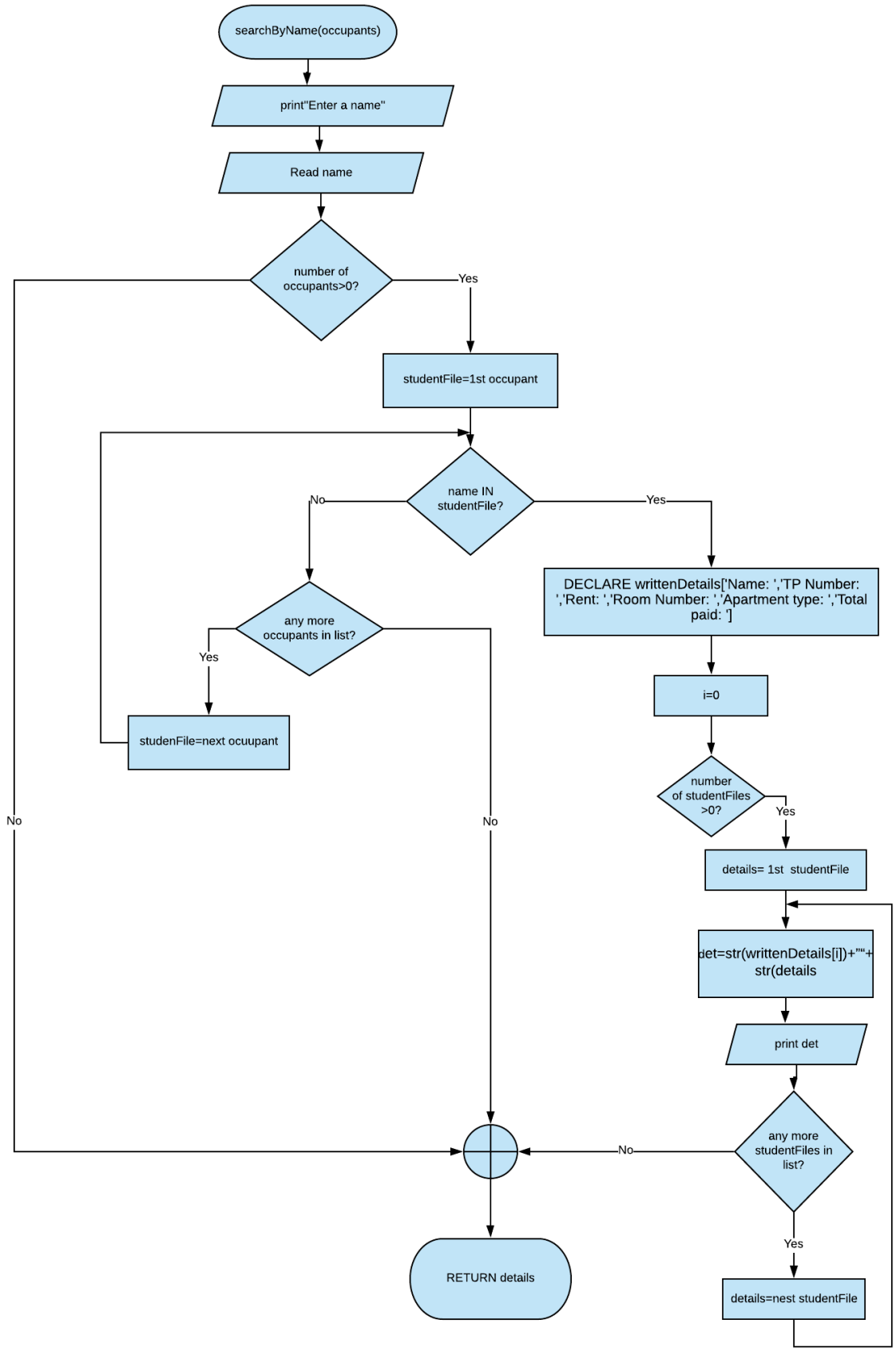
```
def serchByName(occcupants):
    name=input('Enter a name: ')
    for studentFile in occcupants:
        if name.upper() in studentFile :
            writtenDetails=['Name: ','TP Number: ','Rent: ','Room Number: ','apartment type:','Total paid: ']
            i=-1
            for details in studentFile:
                i=i+1
                print(writtenDetails[i],'\t',details)
    return details
```

The program lets the visitor to enter a name to search for it.The program loops through the information of every student in the occpants list till it finds the name in one of the secondry lists. Once the name is found ,the information in this secondary lists gets printed one by one.

B) pseudo code:

```
FUNCTION searchByName(occupants):
    Print''Enter a name''
    Read name
    FOR EACH studentFile in occupants :
        IF name in studentFile THEN
            DECLARE writtenDetails['Name: ','TP Number: ','Rent: ','Room Number:
','Apartment type: ','Total paid: ']
            I=0
            FOR EACH details in studentFile :
                det=str(writtenDetails[i])+"   "+ str(details)
                Print(det)
            ENDFOR
    ENDFOR
    RETURN details
ENDFUNCTION
```

c) Flow chart:


searchByName(occupants)
print"Enter a name"
Read name
number of occupants>0?
Yes
No
studentFile=1st occupant
name IN studentFile?
No
Yes
any more occupants in list?
Yes
No
studenFile=next ocuupant
DECLARE writtenDetails['Name: ','TP Number: ','Rent: ','Room Number: ','Apartment type: ','Total paid: ']
i=0
number of studentFiles >0?
Yes
No
details= 1st studentFile
det=str(writtenDetails[i])+""+ str(details
print det
any more studentFiles in list?
No
Yes
details=nest studentFile
RETURN details

1) Searching by apartment type and room number :

A) Python coding and explanation:

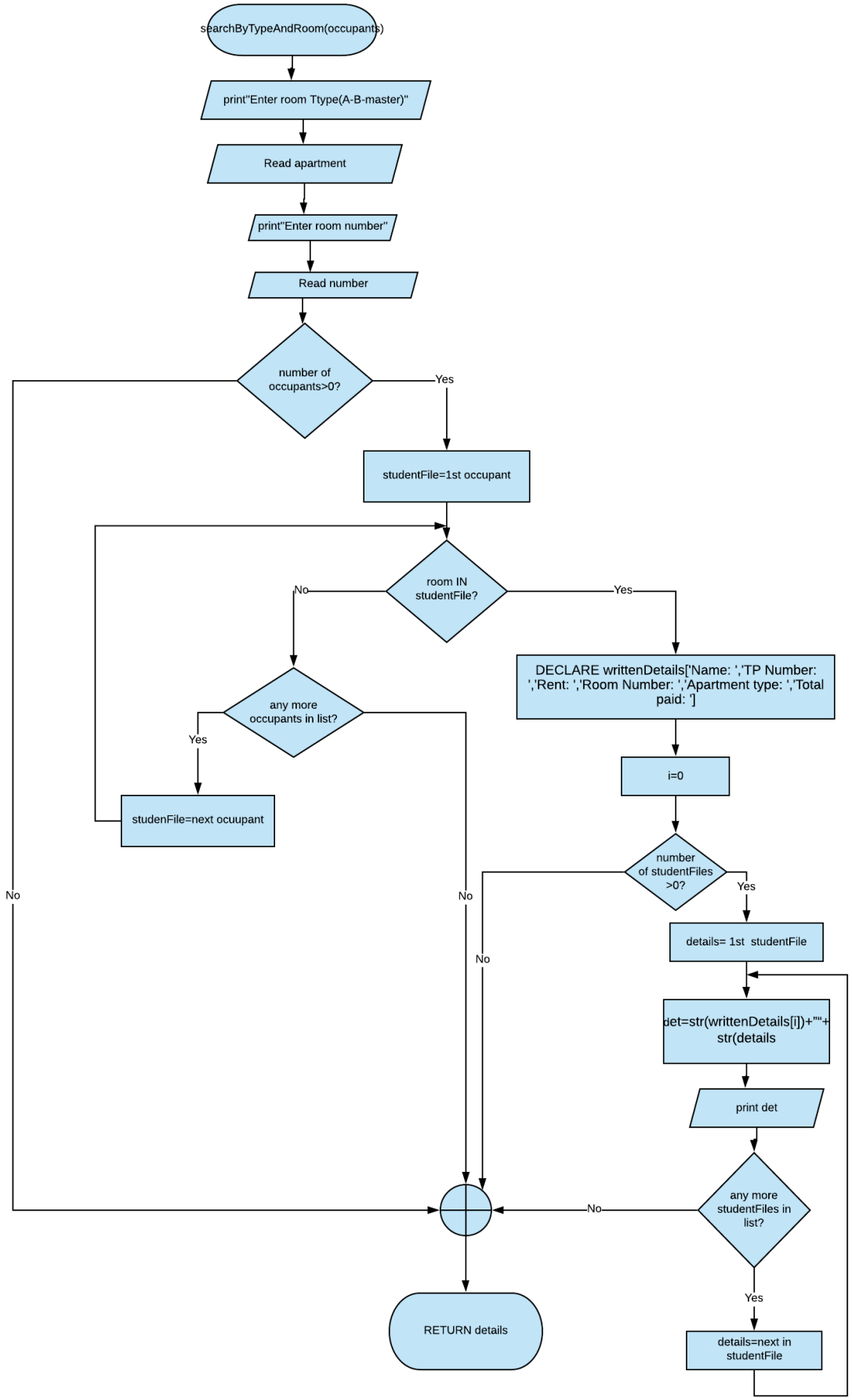
```
def searchByTypeAndRoom(occupants):
    apartment=input('Enter room type[A-B-Master]: ')
    room=input('Enter your room number: ')
    for studentFile in occupants:
        if room in studentFile :
            writtenDetails=['Name: ','TP Number: ','Rent: ','Room Number: ','Apartment type: ','Total paid: ']
            i=-1
            for details in studentFile:
                i=i+1
                print(writtenDetails[i],'\t',details)
    return details
```

Instead of asking for name, the visitor gets asked to enter the room type and number

B) pseudo code:

```
FUNCTION searchByTypeAndRoom(occupants):
        Print''Enter room type [A-B-master] : "
        Read apartment
        Print''Enter room number : "
        Read room
        FOR EACH studentFile in ocupants :
            IF room in studentFile THEN
                DECLARE writtenDetails['Name: ','TP Number: ','Rent: ','Room Number:
','Apartment type: ','Total paid: ']
                i= -1
                FOR EACH details in studentFile :
                    i=i+1
                    Print (writtenDetails[i],''  ",details)
                ENDFOR
            ENDIF
        ENDFOR
       RETURN details
ENDFUNCTION
```

c) Flow chart:


searchByTypeAndRoom(occupants)
print"Enter room Ttype(A-B-master)"
Read apartment
print"Enter room number"
Read number
number of occupants>0?
Yes
No
studentFile=1st occupant
room IN studentFile?
No
Yes
any more occupants in list?
Yes
No
studenFile=next ocuupant
DECLARE writtenDetails['Name: ','TP Number: ','Rent: ','Room Number: ','Apartment type: ','Total paid: ']
i=0
number of studentFiles >0?
Yes
No
details= 1st studentFile
det=str(writtenDetails[i])+""+ str(details
print det
any more studentFiles in list?
No
Yes
details=next in studentFile
RETURN details

THIRD:Accounts information:

1) Printing total deposit collected :

A) Python coding and explanation:

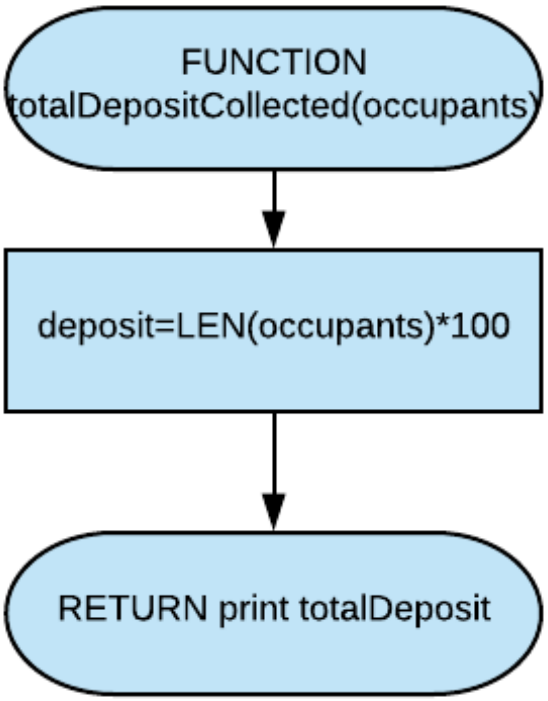
```
def totalDepositCollected(occupants):
    deposit=len(occupants)*100
    return print(deposit)
```

First,the number of occupants gets counted by checking the length of the occupants' list.total deposit equals the number of occupants multiplied by hundered.

B) pseudo code:

```
FUNCTION totalDepositCollected(occupants)
        Deposit=LEN(occupants)*100
        Return print totalDeposit
ENDFUNCTION
```

c) Flow chart:



2) Printing total from students :

A) Python coding and explanation:

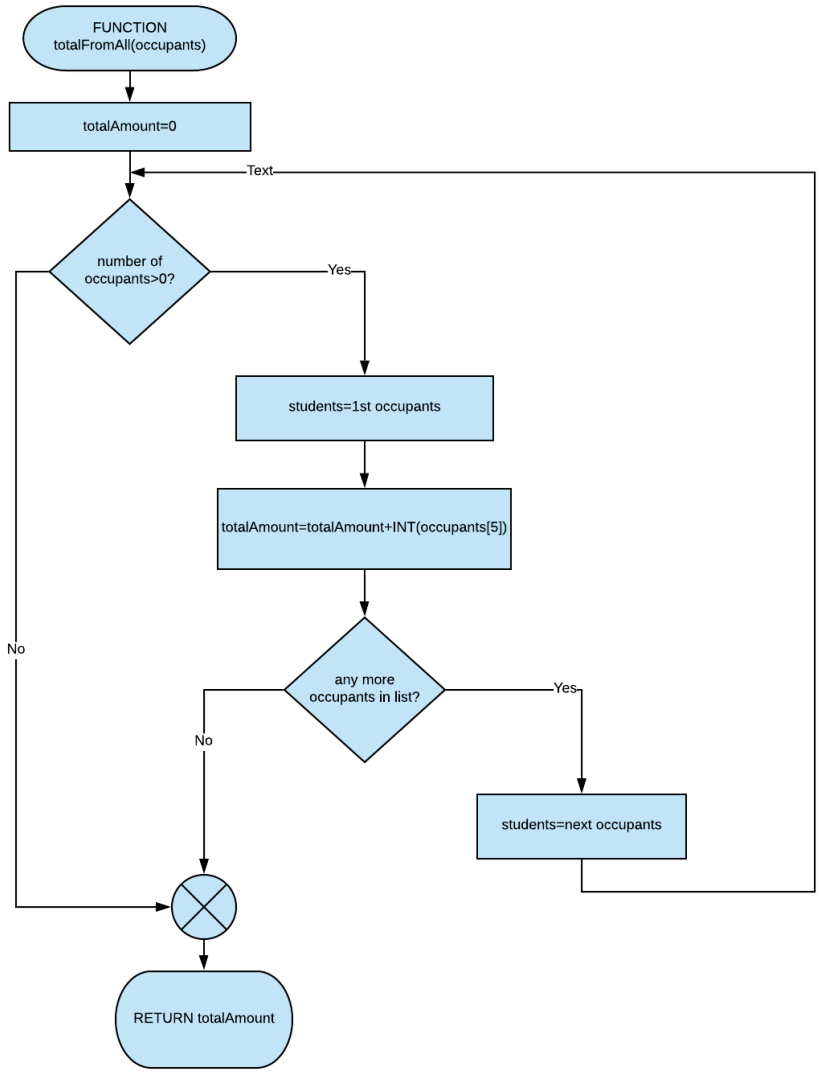
```
def totalFromAll(occupants):
    totalAmount=0
    for students in occupants:
        totalAmount+=int(students[5])
    print(totalAmount)
    return totalAmount
```

Every amount recorded as the fifth element in every secondary list get added together.

B) pseudo code:

```
FUNCTION totalFromAll(occupants):
        totalAmount=0
        FOR EACH students IN occupants:
            totalAmount=totalAmount+INT(students[5])
        END FOR
        Print totalAmount
        RETURN totalAmount
ENDFUNCTION
```

c) Flow chart:



3) Printing total of a student :

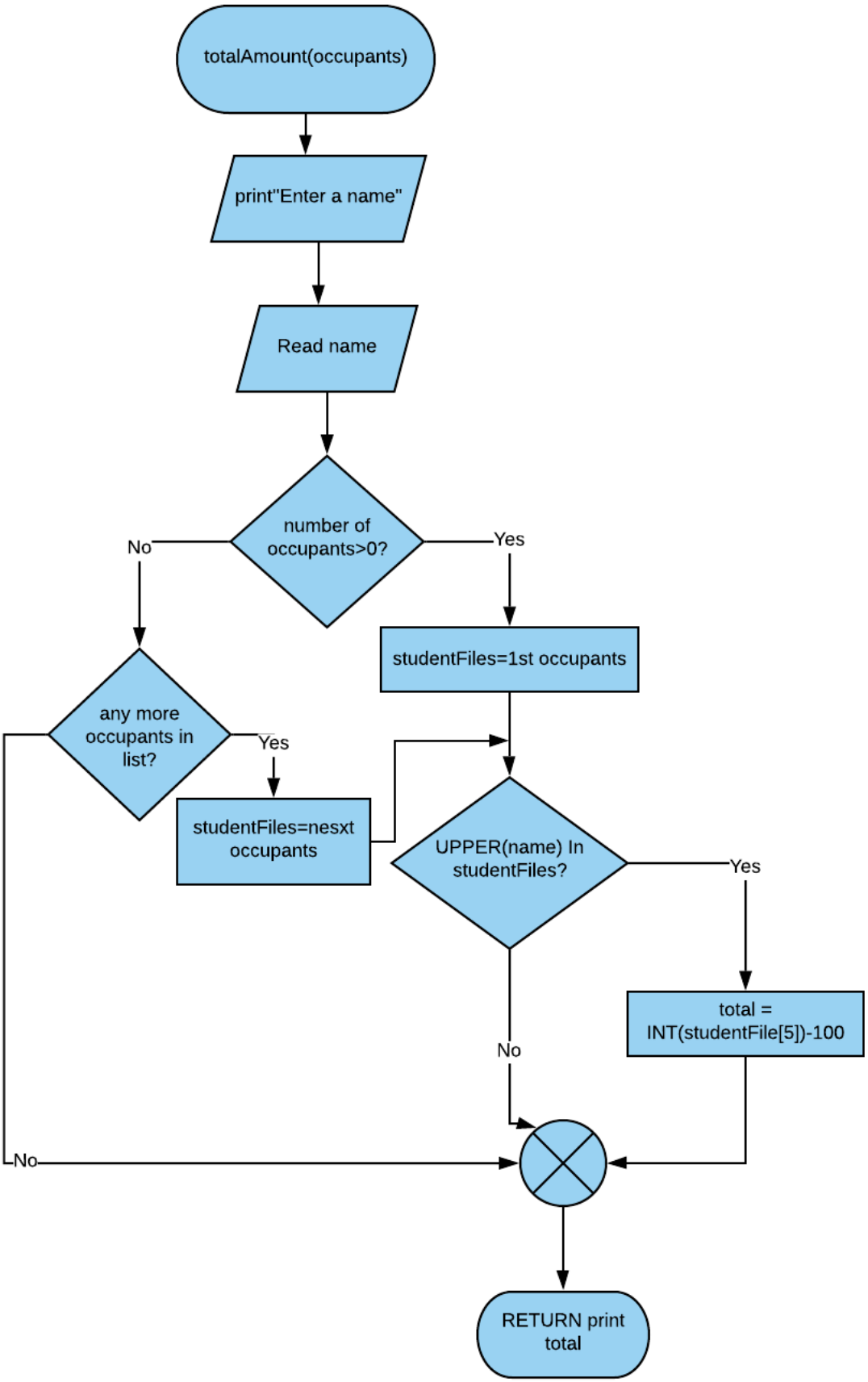
A) Python coding and explanation:

```
    return print(deposit)
def totalAmount(occupants):
    name = input('Enter a name: ')
    for studentFile in occupants:
        if name.upper() in studentFile:
            total=int(studentFile[5])-100
    return print(total)
```

B) pseudo code:

```
FUNCTION totalAmount(occupants)
    Print‘’Enter a name: “
    Read name
    FOR EACH studentFile in occupants
            IF UPPER(name) in studentFile THEN
               Total=INT(studentFile[5])-100
            ENDIF
    ENFFOR
    RETURN Print total
ENDFUNCTION
```

c) Flow chart:


totalAmount(occupants)
print"Enter a name"
Read name
number of occupants>0?
No
Yes
studentFiles=1st occupants
any more occupants in list?
Yes
studentFiles=nesxt occupants
UPPER(name) In studentFiles?
Yes
total = INT(studentFile[5])-100
No
No
RETURN print total

FORTH:Checking out:

A) Python coding and explanation:

```
    return totalAmount
def checkingOut(occupants,A,B,master,name):
    for studentFile in occupants:
        if name.upper() in studentFile:
            if studentFile[4]=='A':
                file=A
            elif studentFile[4]=='B':
                file=B
            else:
                file=master
            file.append(studentFile)
            occupants.remove(studentFile)
            total=int(studentFile[5])-100
            rent=int(studentFile[2])
            months=int(input('How many months have you stayed?: '))
            calc=months*rent
            if total>calc:
                extra=total-calc
                print(extra,'RM will be paid back to you[+100Rm deposit]')
            else:
                extra=calc-total
                print('You are required to pay',extra,'RM [You will get the deposit when you pay the required amount]')
        else:
            print('name not found')
    f = open('details.txt')
    new_list = []
    for line in f:
        line.rstrip()
        new_list.append(line)
        if line.startswith(name):
            new_list.remove(line)
    f.close()
    return new_list
```

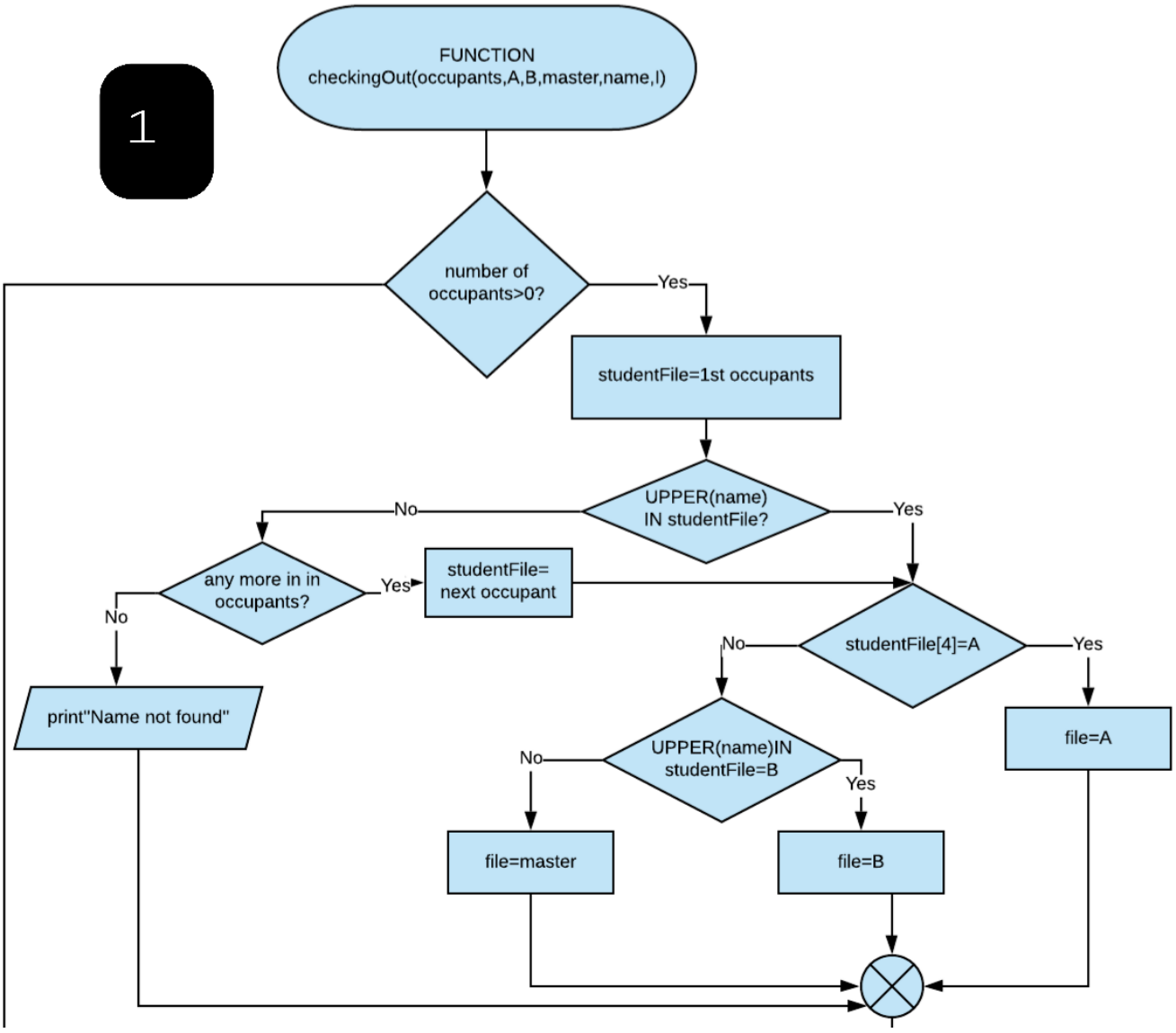
```
                saveInformation(occupants)
            elif choice=='2':
                name=input('Enter a name: ')
                newList=checkingOut(occupants,replacedA,replacedB,replacedMaster,name.upper())
                file = open('details.txt', 'w')
                for dets in newList:
                    file.write(dets)
                    file.write('\n')
                file.close()
```

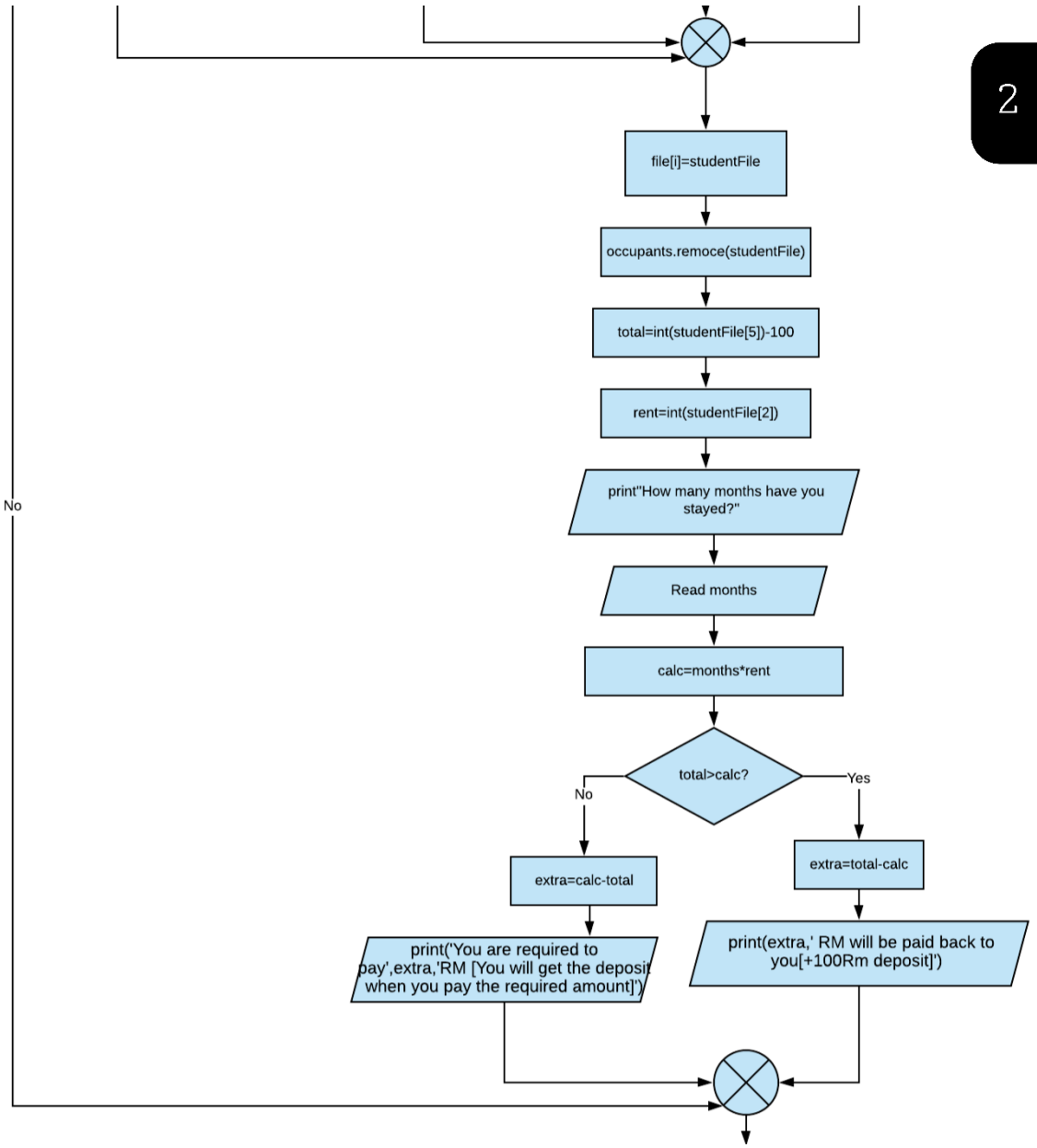
First, the program searches for the name in the occupants file.once it finds the name,the secondary list that contains the checked out occupant gets removed.The total that the checked out paid get assigned to a variable called total.the rent for the months that the occupant stayed gets calculated and gets compared with what he paid.The program then decide if the occupant should be back or if he has an overdue that he needs to pay.the occupants information in the text file get assigned in a list. After removing removing the line that contain the checked out occupant,the information gets re written again in the text file.part of the function had to be separated and put in the menu function because when trying to open the text file two times in a row caused the duplication of the text file instead of assigning the information to the same file.

B) pseudo code:

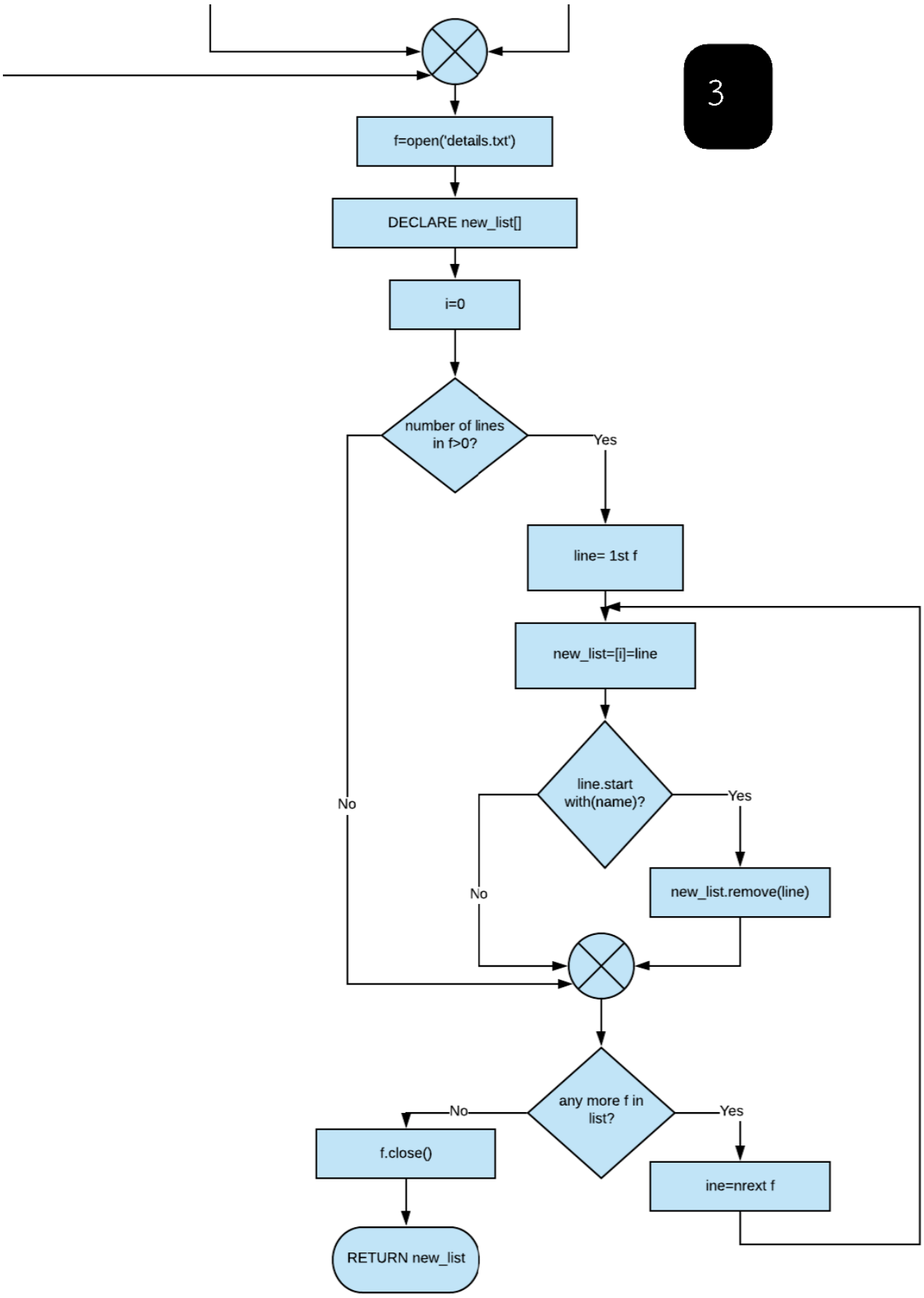
```
FUNCTION checkingOut(occupants,A,B,master,name,i):
   FOR EACH studentFile IN occupants:
      IF UPPER(name) IN studentFile THEN
         IF studentFile[4]=='A' THEN
            file=A
         ELSE:
             IF studentFile[4]=='B' THEN
               file=B
            ELSE:
               file=master
            ENDIF
         ENDIF
         file[i]=studentFile
         occupants.remove(studentFile)
         total=int(studentFile[5])-100
         rent=int(studentFile[2])
         print('How many months have you stayed?: '))
         Read months
         calc=months*rent
         IF total>calc THEN
            extra=total-calc
            print(extra,' RM will be paid back to you[+100Rm deposit]')
         ELSE:
            extra=calc-total
            print('You are required to pay',extra,'RM [You will get the deposit when you
pay the required amount]')
          ENDIF
       ELSE:
          Print''Name not found''
       ENDIF
   ENDFOR
   f = open('details.txt')
   DECLARE new_list  []
   i=0
   FOR EACH line IN f:
      new_list[i]=line
      IF line.startswith(name) THEN
         new_list.remove(line)
      ENDIF
   ENDFOR
   f.close()
   RETURN new_list
ENDFUNCTION
```

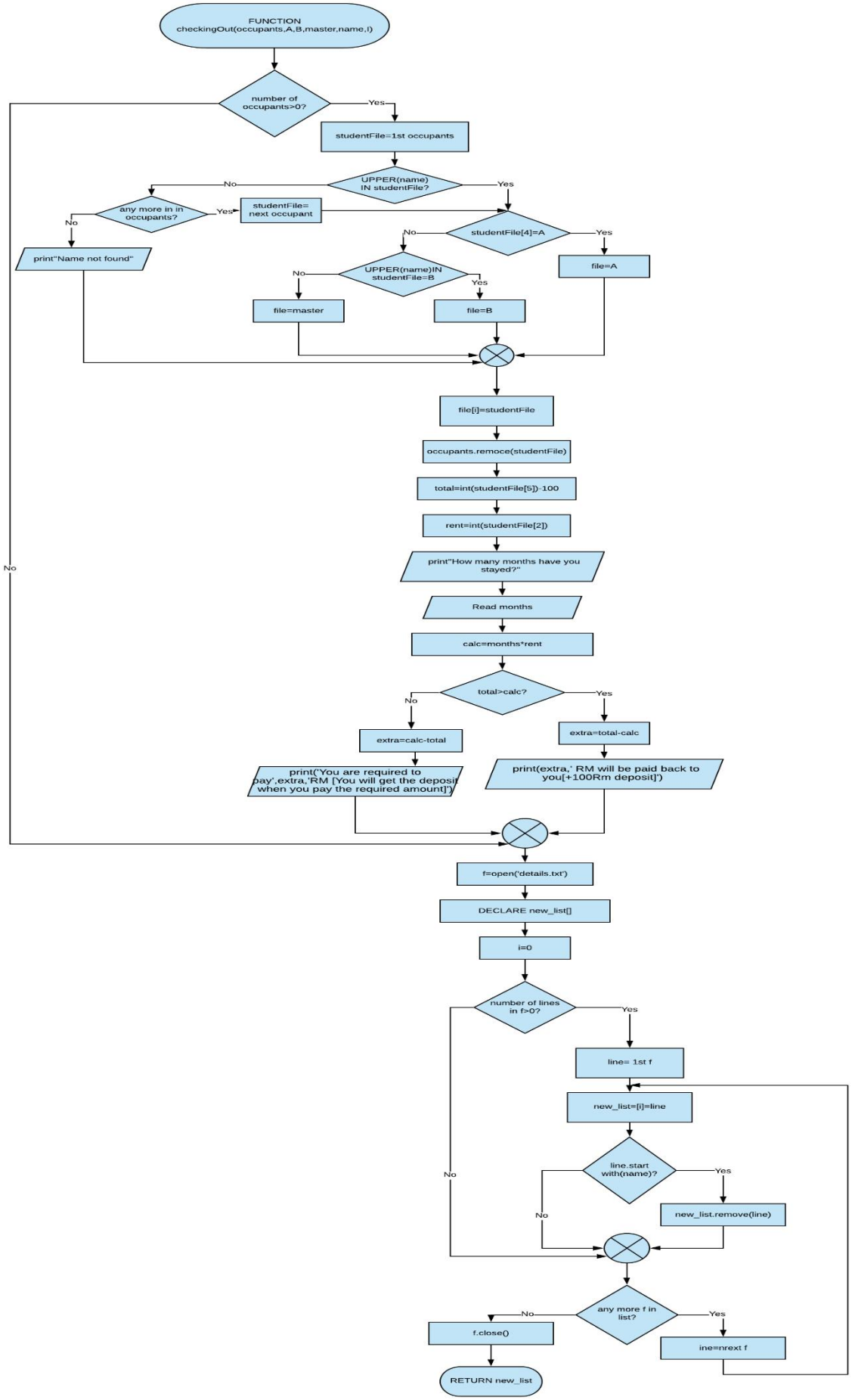
c) Flow chart:


1
FUNCTION checkingOut(occupants,A,B,master,name,l)
number of occupants>0?
Yes
studentFile=1st occupants
UPPER(name) IN studentFile?
No
Yes
any more in in occupants?
Yes
studentFile= next occupant
No
print"Name not found"
No
studentFile[4]=A
Yes
file=A
No
UPPER(name)IN studentFile=B
Yes
file=master
file=B


file[i]=studentFile
occupants.remoce(studentFile)
total=int(studentFile[5])-100
rent=int(studentFile[2])
print"How many months have you stayed?"
Read months
calc=months*rent
total>calc?
No
Yes
extra=calc-total
extra=total-calc
print('You are required to pay',extra,'RM [You will get the deposit when you pay the required amount]')
print(extra,' RM will be paid back to you[+100Rm deposit]')
No

3


f=open('details.txt')
DECLARE new_list[]
i=0
number of lines in f>0?
Yes
No
line= 1st f
new_list=[i]=line
line.start with(name)?
Yes
No
new_list.remove(line)
any more f in list?
No
Yes
f.close()
ine=nrext f
RETURN new_list


FUNCTION checkingOut(occupants,A,B,master,name,l)
number of occupants>0?
Yes
No
studentFile=1st occupants
UPPER(name) IN studentFile?
Yes
No
any more in in occupants?
Yes
No
studentFile= next occupant
print"Name not found"
studentFile[4]=A
Yes
No
file=A
UPPER(name)IN studentFile=B
Yes
No
file=B
file=master
file[i]=studentFile
occupants.remoce(studentFile)
total=int(studentFile[5])-100
rent=int(studentFile[2])
print"How many months have you stayed?"
Read months
calc=months*rent
total>calc?
Yes
No
extra=total-calc
extra=calc-total
print(extra,' RM will be paid back to you[+100Rm deposit]')
print('You are required to pay',extra,'RM ['You will get the deposit when you pay the required amount]')
No
f=open('details.txt')
DECLARE new_list[]
i=0
number of lines in f>0?
Yes
line= 1st f
new_list=[i]=line
line.start with(name)?
Yes
No
new_list.remove(line)
No
any more f in list?
Yes
No
f.close()
ine=nrext f
RETURN new_list

FIFTH:Menu:

A) Python coding and explanation:

```
def menu():
    occupants = []
    typeA = []
    typeB = []
    masterRooms = []
    replacedA=[]
    replacedB=[]
    replacedMaster=[]
    i=1
    while i==1:
        print('Select the operation you want to perform: ')
        print('1.For housing registration ')
        print('2.For checking out')
        print('3.For printing total deposit')
        print('4.For printing total amount excluding the deposit')
        print('5.For printing Total amount received from students')
        print('6.For searching by name')
        print('7.For searching by room type and room number ')
        print('8.To exit')
        choice = input('Enter your choice: ')
        if choice=='1':
            studentRegistration(occupants,typeA,typeB,masterRooms,replacedA,replacedB,replacedMaster)
            saveInformation(occupants)
        elif choice=='2':
            name=input('Enter a name: ')
            newList=checkingOut(occupants,replacedA,replacedB,replacedMaster,name.upper())
            file = open('details.txt', 'w')
            for dets in newList:
                file.write(dets)
                file.write('\n')
            file.close()
        elif choice=='3':
            totalDepositCollected(occupants)
        elif choice=='4':
            totalAmount(occupants)
        elif choice=='5':
            totalFromAll(occupants)
        elif choice=='6':
            serchByName(occupants)
        elif choice=='7':
            searchByTypeAndRoom(occupants)
        elif choice=='8':
            i=2
            continue
        else:
            print('Wrong choice')
            continue
```

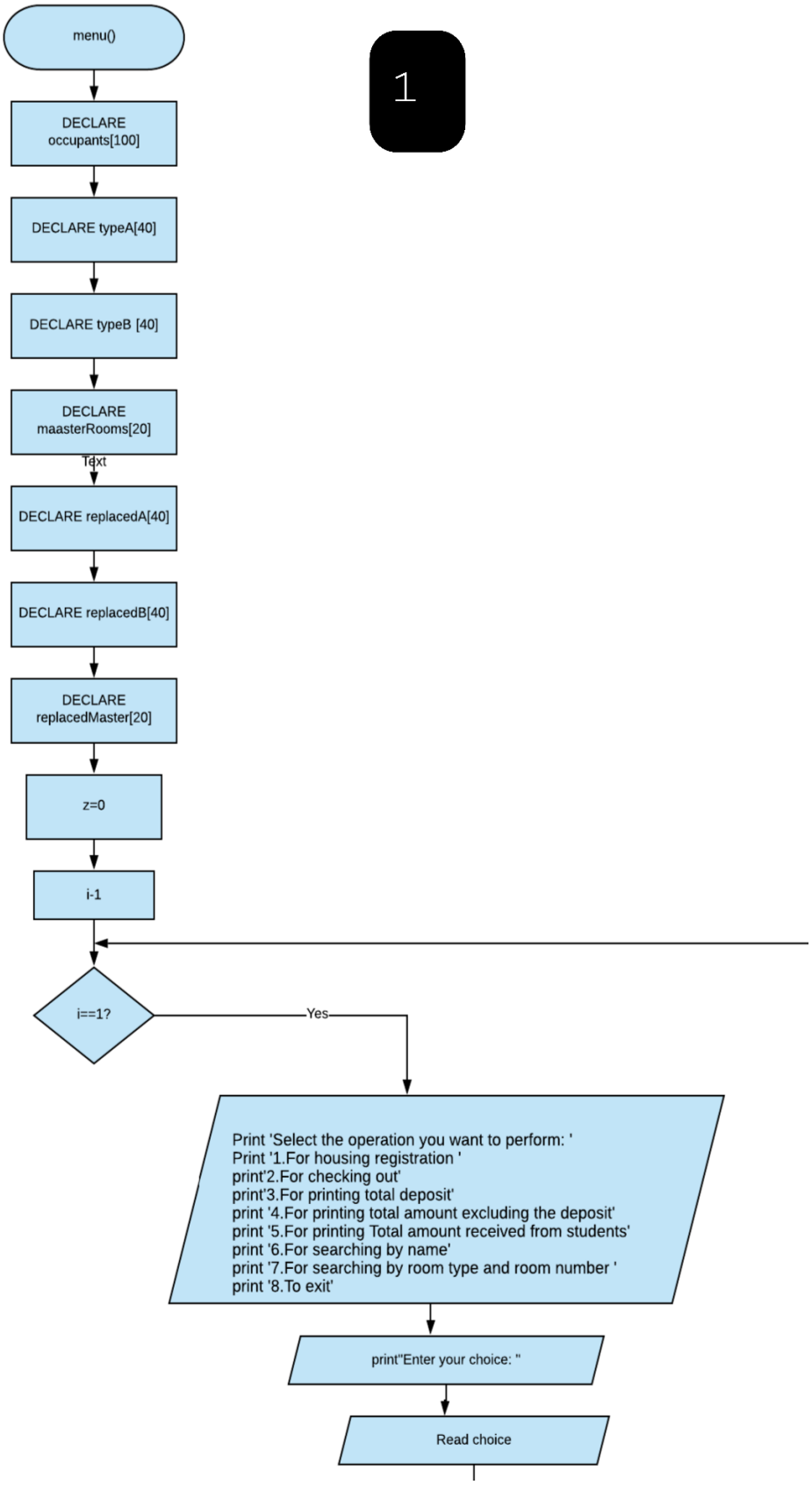
A menu of different function is displayed for visitor to choose which operation he wants to perform.All lists that are used in functions are declared here.

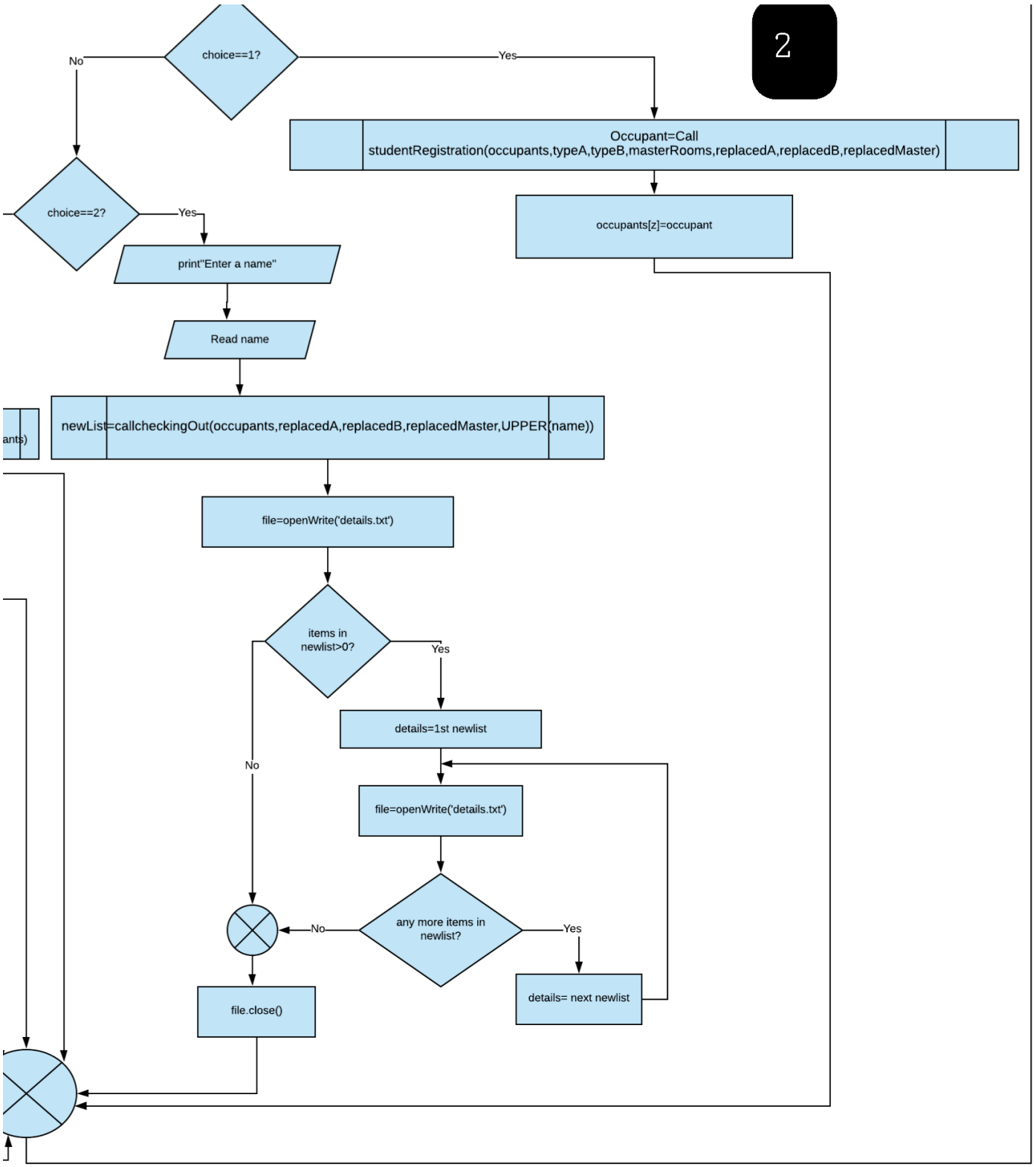
B) pseudo code:

```
FUNCTION menu()
  DECLARE occupants [100]
  DECLARE typeA[40]
  DECLARE typeB[40]
  DECLARE masterRooms[20]
  DECLARE replacedA[40]
  DECLARE replacedB[40]
  DECLARE replacedMaster[20]
  Z=0
  I=1
  DOWHILE I ==1
    Print 'Select the operation you want to perform: '
    Print '1.For housing registration '
    print'2.For checking out'
    print'3.For printing total deposit'
    print '4.For printing total amount excluding the deposit'
    print '5.For printing Total amount received from students'
    print '6.For searching by name'
    print '7.For searching by room type and room number '
    print '8.To exit'
    Print ’Enter your choice: ‘
    Read choice
    IF choice==1 THEN
      Occupant=Call
studentRegistration(occupants,typeA,typeB,masterRooms,replacedA,replacedB,replacedM
aster)
      Occupants[z]=occupant
    ELSE:
      IF choice==2 THEN
        Print’Enter a name :’
        Read name
        newList=call
checkingOut(occupants,replacedA,replacedB,replacedMaster,UPPER(name))
        file=opemWrite(‘details.txt’)
        FOR EACH dets IN newList:
          File.writeLine(dets)
        ENDFOR
        File.close()
      ELSE:
        IF choice==3 THEN
          Call totalDepositCollected(occupants)
        ELSE:
          IF choice==4 THEN
            Call totalAmount(occupants)
          ELSE:
            IF choice==5 THEN
              Call totalFromAll(occupants)
            ELSE:
```

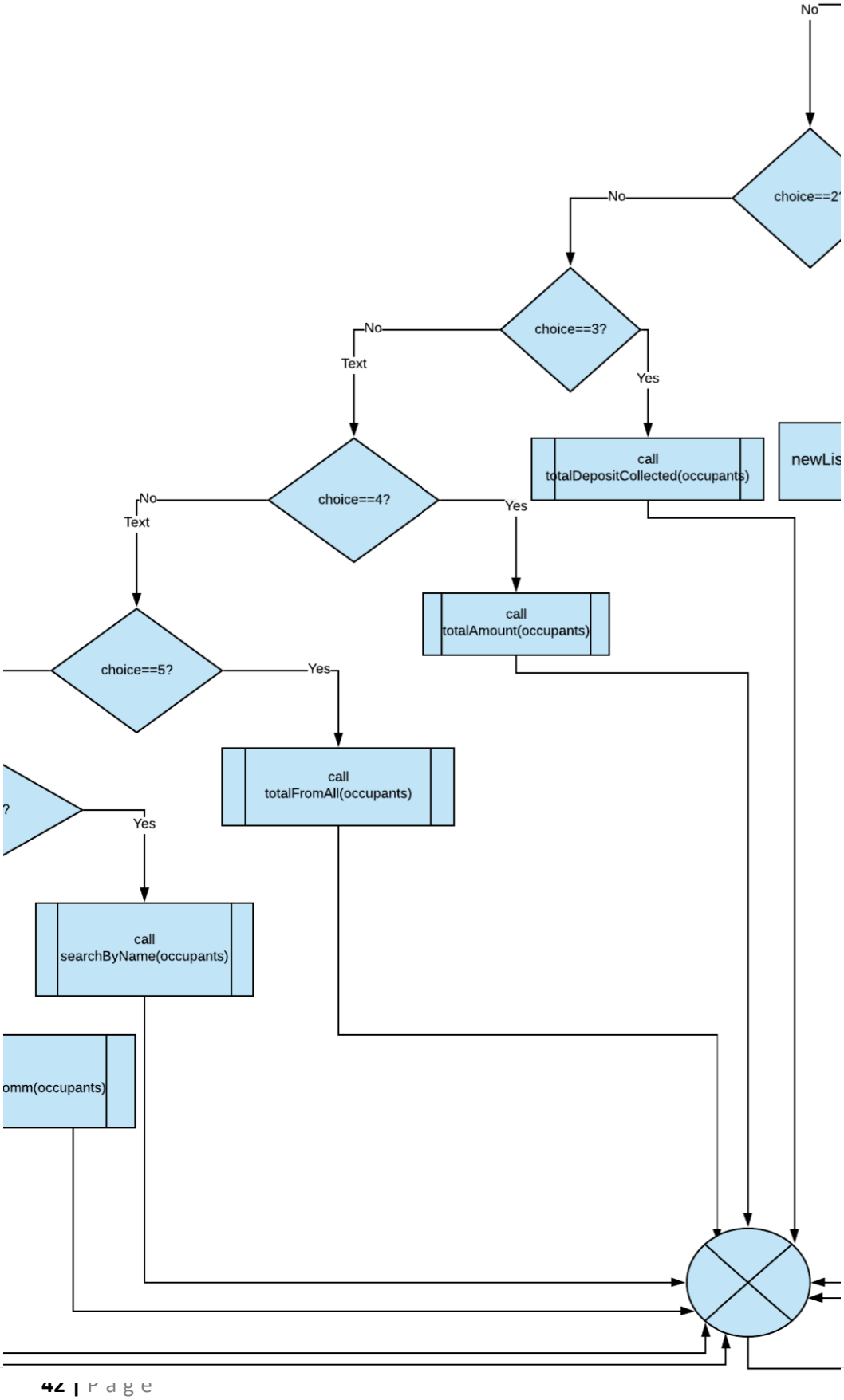
```
            IF choice==6 THEN
              Call searchByname(occupants)
            ELSE:
              IF choice==7 THEN
                Call searchByTypeAndRoom(occupants)
              ELSE:
                IF choice==8 THEN
                  I=2
                ELSE:
                  Print'Invalid input'
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
 ENDDO
 RETURN choice
END FUNCTION
```

c) Flow chart:

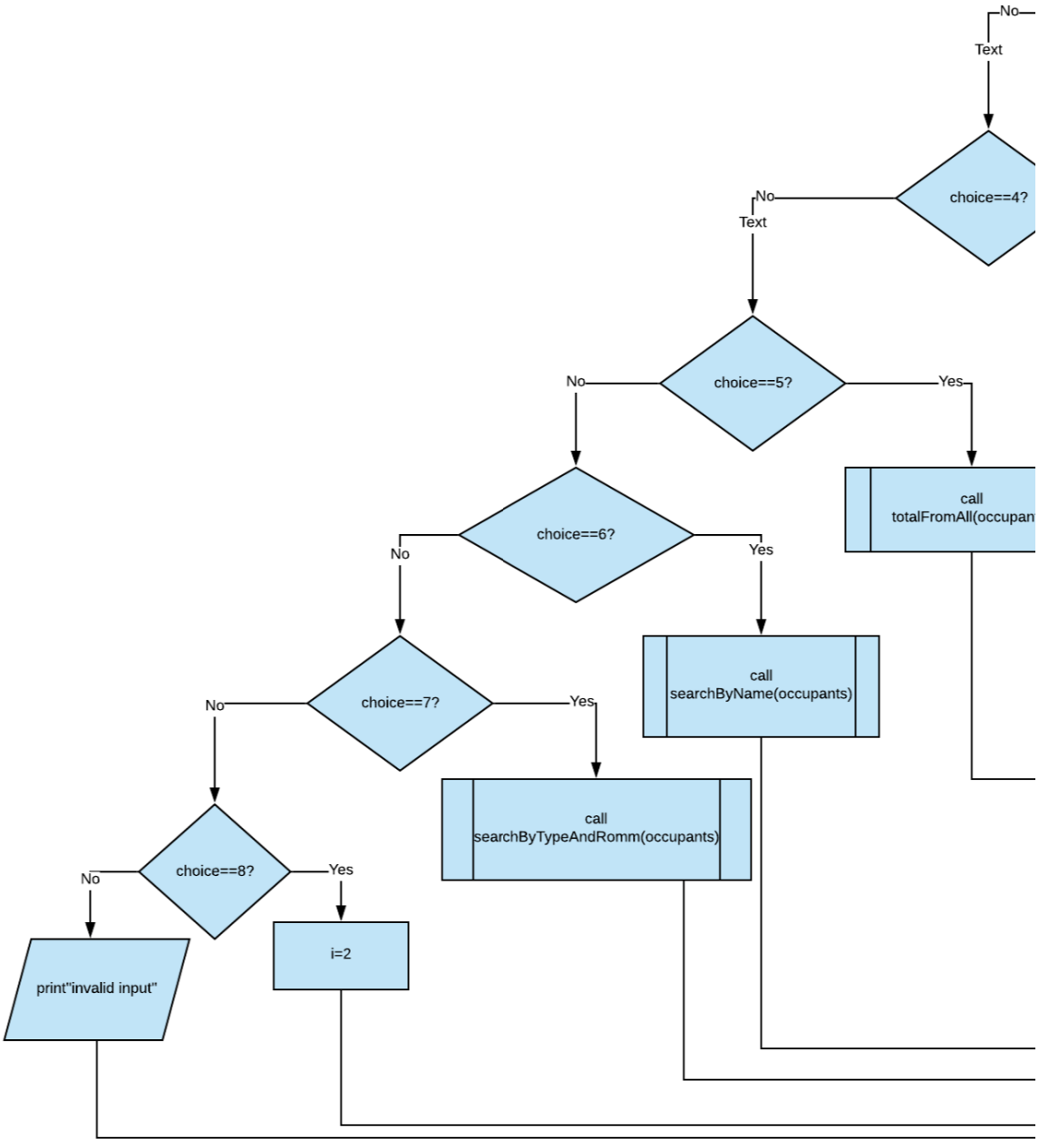

menu()
1
DECLARE occupants[100]
DECLARE typeA[40]
DECLARE typeB [40]
DECLARE maasterRooms[20]
Text
DECLARE replacedA[40]
DECLARE replacedB[40]
DECLARE replacedMaster[20]
z=0
i-1
i==1?
Yes
Print 'Select the operation you want to perform: '
Print '1.For housing registration '
print'2.For checking out'
print'3.For printing total deposit'
print '4.For printing total amount excluding the deposit'
print '5.For printing Total amount received from students'
print '6.For searching by name'
print '7.For searching by room type and room number '
print '8.To exit'
print"Enter your choice: "
Read choice

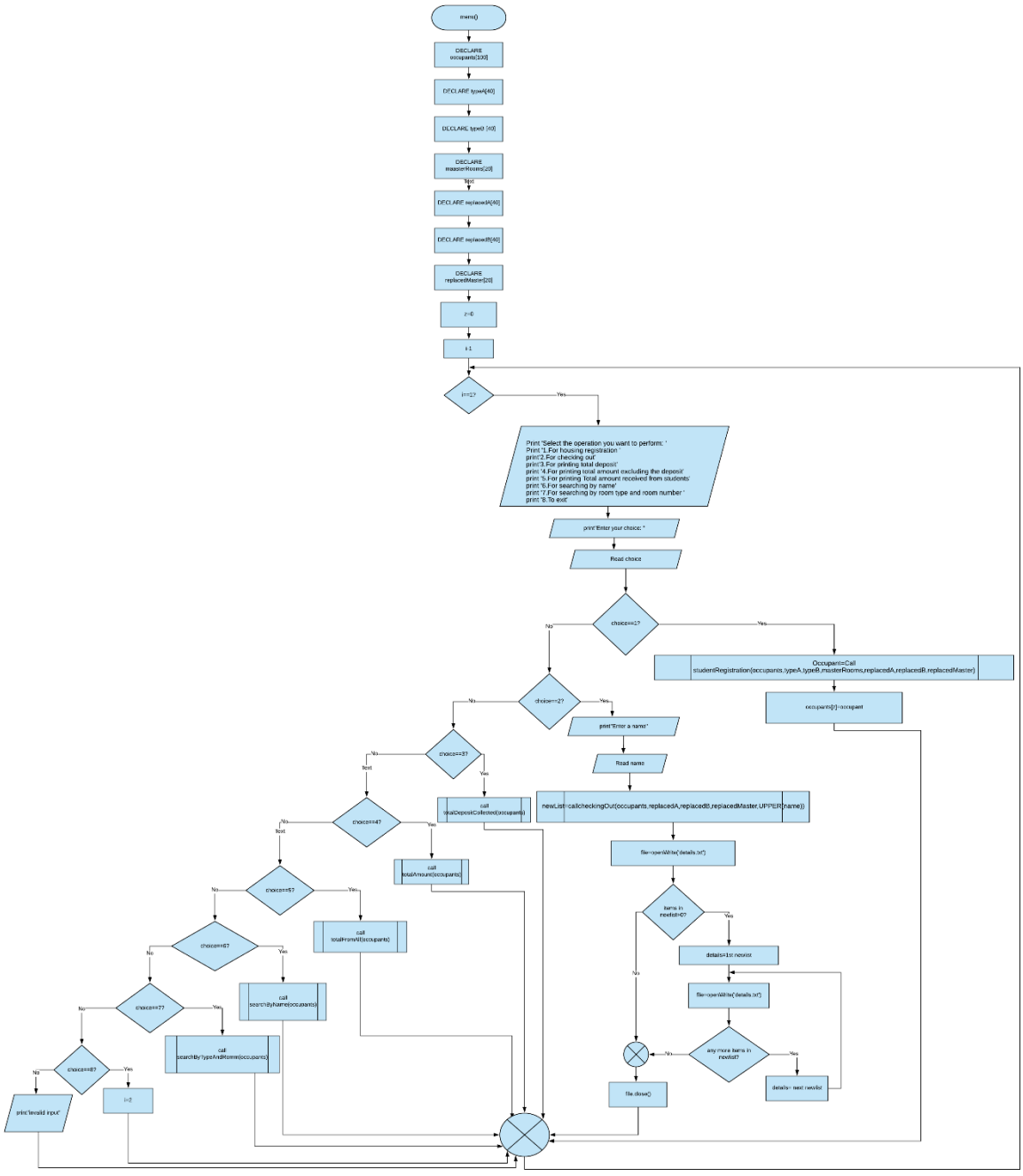

2
choice==1?
No
Yes
Occupant=Call studentRegistration(occupants,typeA,typeB,masterRooms,replacedA,replacedB,replacedMaster)
choice==2?
Yes
occupants[z]=occupant
print"Enter a name"
Read name
ants)
newList=callcheckingOut(occupants,replacedA,replacedB,replacedMaster,UPPER(name))
file=openWrite('details.txt')
items in newlist>0?
Yes
details=1st newlist
No
file=openWrite('details.txt')
any more items in newlist?
No
Yes
details= next newlist
file.close()

3


No
choice==2
No
choice==3?
No
Text
Yes
call
totalDepositCollected(occupants)
newLis
choice==4?
No
Text
Yes
call
totalAmount(occupants)
choice==5?
Yes
call
totalFromAll(occupants)
Yes
call
searchByName(occupants)
omm(occupants)

4


No
Text
choice==4?
No
Text
choice==5?
No
Yes
choice==6?
No
Yes
choice==7?
No
Yes
choice==8?
No
Yes
print"invalid input"
i=2
call searchByTypeAndRomm(occupants)
call searchByName(occupants)
call totalFromAll(occupan


menu()
DECLARE occupants[100]
DECLARE typeA[40]
DECLARE typeB [40]
DECLARE masterRooms[20]
type
DECLARE replacedA[40]
DECLARE replacedB[40]
DECLARE replacedMaster[20]
z=0
i=1
i==7
Yes
Print 'Select the operation you want to perform: '
Print '1.For housing registration
print'2.For checking out
print'3.For printing total deposit'
print '4.For printing total amount excluding the deposit'
print '5.For printing total amount received from students'
print '6.For searching by name'
print '7.For searching by room type and room number '
print '8.To exit'
print'Enter your choice: '
Read choice
choice==1?
No
Yes
Occupant=Call studentRegistration(occupants,typeA,typeB,masterRooms,replacedA,replacedB,replacedMaster)
occupants[z]=occupant
choice==2?
print"Enter a name"
Read name
choice==3?
call totalDepositCollected(occupants)
newList=callcheckingOut(occupants,replacedA,replacedB,replacedMaster,UPPER(name))
file=openWrite('details.txt')
choice==4?
call totalAmount(occupants)
items in newlist>0?
details=1st newlist
choice==5?
call totalFromAll(occupants)
file=openWrite('details.txt')
choice==6?
call searchByName(occupants)
any more items in newlist?
details= next newlist
choice==7?
call searchByTypeAndRoom(occupants)
file.close()
choice==8?
print'Invalid input'
i=2

RUNNIG THE CODE:

```
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: raof
Enter your TP number: tp01
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: no
Your total is:  2100 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :partial
You are required to pay at least 50% of total rent + 100 RM deposit
Enter the amount you are going to pay: 1600
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: zane
Enter your TP number: tp02
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: yes
Your total is:  2350 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :full
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
```

```
Enter your choice: 1
Enter your name: fahd
Enter your TP number: tp03
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: no
Your total is:  2100 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :partial
You are required to pay at least 50% of total rent + 100 RM deposit
Enter the amount you are going to pay: 1700
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: ramy
Enter your TP number: tp04
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: no
Your total is:  2100 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :partial
You are required to pay at least 50% of total rent + 100 RM deposit
Enter the amount you are going to pay: 1900
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: ahmd
Enter your TP number: tp05
```

```
Enter your choice: 1
Enter your name: zena
Enter your TP number: tp06
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: yes
Your total is:  2350 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :full
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: rily
Enter your TP number: tp07
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
NOTICEE: All types have kitchen and laundry room
Enter your choice(A or B): a
Do you want internet connection (50 RM) [yes or no]: yes
Your total is:  2350 [Notice: 100 RM deposit included]
Are you going to pay full or partial? :partial
You are required to pay at least 50% of total rent + 100 RM deposit
Enter the amount you are going to pay: 1850
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 1
Enter your name: seif
Enter your TP number: tp08
Apartment types:
1)  Type A:2 single rooms (400 RM)
2)  Type B:2 single rooms (300 RM) - 1 Master bedroom (500 RM)
```

```
apartment type:          A
Total paid:      2350
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 7
Enter room type[A-B-Master]: a
Enter your room number: a-3-2
Name:    ZENA
TP Number:       tp06
Rent:    400
Room Number:     a-3-2
Apartment type:          A
Total paid:      2350
Select the operation you want to perform:
1.For housing registration
2.For checking out
3.For printing total deposit
4.For printing total amount excluding the deposit
5.For printing Total amount received from students
6.For searching by name
7.For searching by room type and room number
8.To exit
Enter your choice: 6
Enter a name: ahmd
Name:    AHMD
TP Number:       tp05
Rent:    400
Room Number:     a-3-1
apartment type:          A
Total paid:      2100
```

TEXT FILE:

File Edit Format View Help

ZANE	tp02	400	a-1-2	A	2350
FAHD	tp03	400	a-2-1	A	1700
RAMY	tp04	400	a-2-2	A	1900
AHMD	tp05	400	a-3-1	A	2100
ZENA	tp06	400	a-3-2	A	2350
RILY	tp07	400	a-4-1	A	1850

WENDY	tp12	500	master-3	MASTER	2600
REMY	tp13	500	master-4	MASTER	1450
RUBY	tp14	300	b-1-1	B	1800
RIM	tp15	300	b-1-2	B	1600
SED	tp16	300	b-2-1	B	1600