



COMPUTE SCIENCE

EDEXCEL INTERNATIONAL GCSE
O LEVEL

PYTHON ACTIVITIES
CHAPTE 1 – CHAPTER 29

Dr. Saw Myat Sandar
STUDENT WORK BOOK

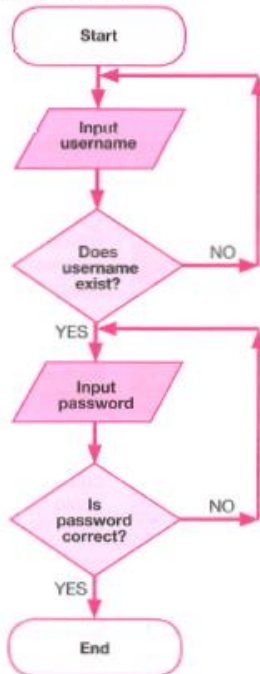
Python in Other Chapters

1.Chapter1 Activity6

ACTIVITY 6

WRITTEN DESCRIPTIONS OF ALGORITHMS

This algorithm is displayed as a flowchart.



▲ Figure 1.5 Flowchart of an algorithm

Produce a written description of this algorithm.

Ask the user to enter their username.

Repeat until an existing username is entered.

Next ask the user to enter their password.

Repeat until the correct password is entered.

```

# Assuming you have a list of existing usernames and their
corresponding passwords
#existing_usernames = {'user1': 'password1', 'user2': 'password2',
'user3': 'password3'}
existing_usernames = ['user1','pw1']
# Ask the user to enter their username
while True:
    username = input("Enter your username: ")
    if username in existing_usernames:
        break
    else:
        print("Username not found. Please try again.")
# Ask the user to enter their password
while True:
    password = input("Enter your password: ")
    if password == existing_usernames[username]:
        break
    else:
        print("Incorrect password. Please try again.")
# Authentication successful
print("Login successful. Welcome,", username)
  
```



2.Chapter1 C1

C1 Develop an algorithm using a flowchart that asks the user to enter their height (in metres) and weight (in kilograms) and displays their body mass index (BMI). The formula for calculating BMI is $\text{weight}/\text{height}^2$.

```
SEND 'Enter your height (in inches)' TO DISPLAY
RECEIVE height FROM KEYBOARD
SEND 'Enter your weight (in lb)' TO DISPLAY
RECEIVE weight FROM KEYBOARD
SET bmi TO weight/height^2
SEND bmi TO DISPLAY
```

```
# Receive height from the user
height = float(input("Enter your height (in inches):"))

# Receive weight from the user
weight = float(input("Enter your weight (in lb):"))

# Calculate BMI (Body Mass Index)
bmi = 703*weight / (height ** 2)
# Display the calculated BMI
print("Your BMI is:", bmi)
```



3.Chapter2 Activity 8

Calculating Grades - A school uses this algorithm to calculate the grade that students achieve in end-of-topic tests.

RECEIVE testScore FROM KEYBOARD

IF testScore >=80 THEN

 SEND 'A' TO DISPLAY

ELSE

 IF testScore >=70 THEN

 SEND 'B' TO DISPLAY

 ELSE

 IF testScore >=60 THEN

 SEND 'C' TO DISPLAY

 ELSE

 IF testScore >0 THEN

 SEND 'D' TO DISPLAY

 ELSE

 SEND 'FAIL' TO DISPLAY

 END IF

 END IF

 END IF

END IF

```
# Assuming testScore is a variable that holds the student's test score
```

```
testScore = float(input("Enter the test score: "))
```

```
# Determine the grade based on the test score
```

```
if testScore >= 80:
```

```
    grade = 'A'
```

```
elif testScore >= 70:
```

```
    grade = 'B'
```

```
elif testScore >= 60:
```

```
    grade = 'C'
```

```
elif testScore > 0:
```

```
    grade = 'D'
```

```
else:
```

```
    grade = 'FAIL'
```

```
# Display the calculated grade
```

```
print("Grade:", grade)
```



4.Chapter2 C2

C2 Develop an algorithm expressed as a flowchart to control the heating in a house. A thermostat monitors the temperature within the house. During the week the temperature should be 20°C between 06.00 and 08.30 in the morning and between 17.30 and 22.00 at night. At weekends it should be 22°C between 08.00 and 23.00. If the temperature in the house falls below 10°C at any time the boiler is switched on.

```
import datetime
# Get current date and time
current_time = datetime.datetime.now()
current_day = int(input("Enter day (0 to 6:))) # Monday is 0 and Sunday is 6
current_hour = int(input("Enter hour:"))
#current_time = datetime.datetime.now()
#current_day = current_time.weekday() # Monday is 0 and Sunday is 6
#current_hour = current_time.hour

print(type(current_time))
print(type(current_day))
print(type(current_hour))

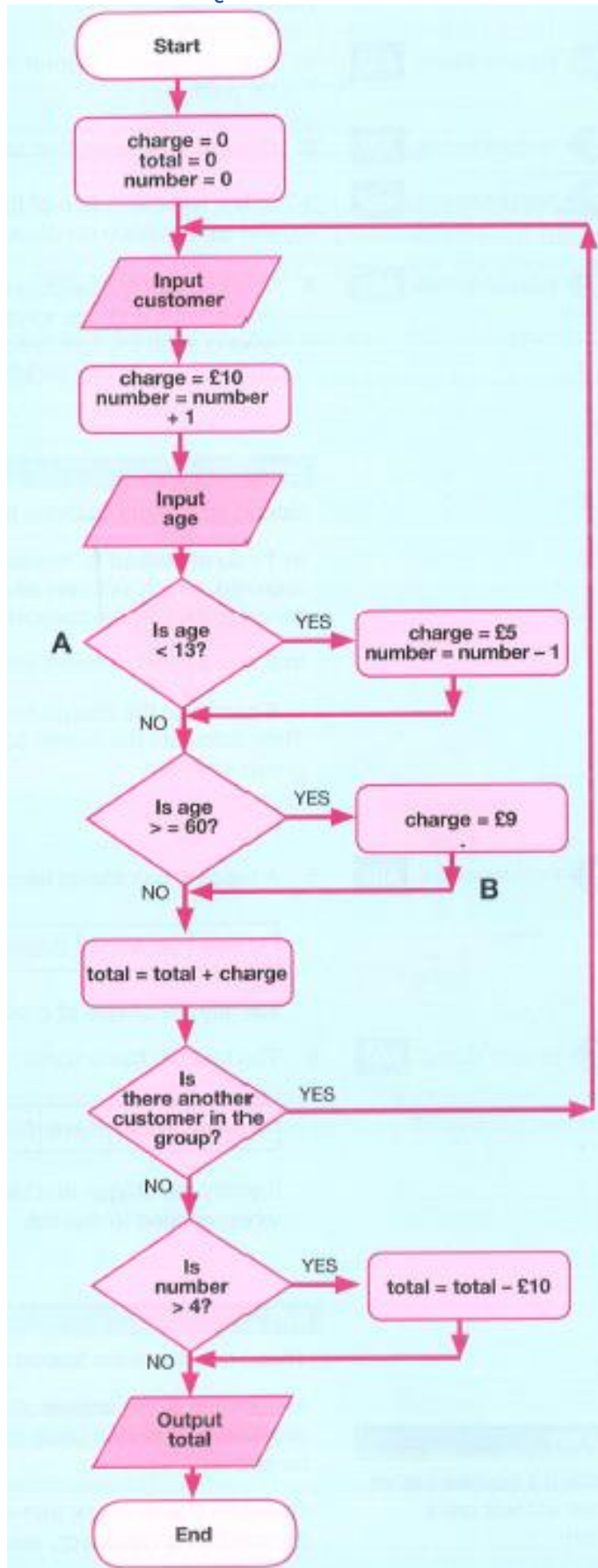
# Weekday temperature conditions
if 0 <= current_day <= 4: # Monday to Friday
    if (6 <= current_hour <= 8.5) or (17.5 <= current_hour <= 22):
        temperature = 20
    else:
        temperature = None
else: # Weekend temperature conditions
    if 8 <= current_hour <= 23:
        temperature = 22
    else:
        temperature = None

# Check for minimum temperature
if temperature is not None and temperature < 10:
    # Switch on boiler
    print("Boiler switched on!")

# Adjust temperature (this is where you would set the actual temperature in your heating system)
if temperature is not None:
    print(f"Temperature adjusted to {temperature}°C")
```



5.Unit1 Unit Question 1



```
charge=0
total=0
number=0
anotherGo="y"
while anotherGo=="Y" or anotherGo=="y":
    charge=10
    number=number+1
    age=int(input("Enter customer age:"))

    if age<13:
        charge=5
        number=number-1
    elif age>=60:
        charge=9
    total=total+charge
    print(total)
    anotherGo=input("Another member y or n:")

if number>4:
    total=total-10
    print(total)
else:
    print(total)
```

```
Shell x
>>> %Run ch1-unitquestion.py

Enter customer age:8
Another member y or n:y
Enter customer age:12
Another member y or n:y
Enter customer age:35
Another member y or n:y
Enter customer age:32
Another member y or n:y
Enter customer age:65
Another member y or n:n
39
```

```
Shell x
>>> %Run ch1-unitquestion.py

Enter customer age:8
Another member y or n:y
Enter customer age:15
Another member y or n:y
Enter customer age:35
Another member y or n:y
Enter customer age:33
Another member y or n:y
Enter customer age:65
Another member y or n:n
44
```



6.Unit1 Unit Question 7

7 Create an algorithm to calculate the cost of sending a parcel.
If the weight of the parcel is 2 kg or under then the standard charge is \$3.

There is then a charge of \$2 for each extra kilogram up to 10 kg.
After 10 kg the charge per extra kilogram is \$3.

a Display your algorithm as a flowchart.

b Construct your algorithm as pseudocode.

c Write your program using high level language.

```
SEND "Enter the parcel weight" TO DISPLAY
RECEIVE weight FROM KEYBOARD
IF (weight<=2) THEN
    SET cost TO 3
ELSE
    IF (weight>10) THEN
        SET cost TO 3+(8*2)+(weight-10)*3
    ELSE
        SET cost TO 3 + (weight - 2) * 2
    END IF
END IF
SEND cost TO DISPLAY
```

```
# Display prompt and receive input from the user
weight = int(input("Enter the parcel weight (in kg):"))

# Calculate shipping cost based on the provided conditions
if weight <= 2:
    cost = 3
else:
    if weight > 10:
        cost = 3 + (8 * 2) + ((weight - 10) * 3)
    else:
        cost = 3 + ((weight - 2) * 2)

# Display the calculated cost
print("Delivery cost:", cost)
```




```
SET parcel TO "y"
WHILE parcel = "y" DO # The loop will run while parcel = "y"
    RECEIVE weight FROM (INTEGER) KEYBOARD
    IF weight <= 2 THEN
        SET cost TO 3 # Parcels up to 2kg cost $2
    ELSE
        IF weight > 10 THEN
            SET cost To 3 + (8*2) + ((weight - 10) * 3)
        ELSE
            SET cost TO 3 + ((weight -2)*2)
        END IF
    END IF
END IF
SEND cost TO PRINTER
SEND 'Press "y" to process another parcel.' TO DISPLAY
RECEIVE parcel FROM (STRING) KEYBOARD
END WHILE
```

```
parcel = "y"

while parcel.lower() == "y":
    # Receive weight from the user
    print("Enter the parcel weight (in kg):")
    weight = float(input())

    # Calculate shipping cost based on the provided conditions
    if weight <= 2:
        cost = 3
    else:
        if weight > 10:
            cost = 3 + (8 * 2) + ((weight - 10) * 3)
        else:
            cost = 3 + ((weight - 2) * 2)

    # Display the calculated cost
    print("Shipping cost:", cost)

    # Prompt the user to process another parcel
    print('Press "y" to process another parcel.')
    parcel = input()

# End of the loop
print("Parcel processing complete.")
```



7.Chapter 13 Activity 10

Encryption

- 1 Create an algorithm that will encrypt the following sentence:
' The ASCII code represents characters.'
- 2 Encrypt it with a shift of 3 to the right (i.e. A should be encrypted as D).
(Remember: the spaces and the full stop should not be changed.)
- 3 Output the encrypted text.
- 4 Present your algorithm as pseudocode or code and test it in the programming language you are studying.

```

message=input("Enter the message to encrypt:")
shift=int(input("Enter the size of the shift:"))
secreteMessage=""
for character in message:
    number=ord(character)#ord('a')=97
    if character.lower() in 'abcdefghijklmnopqrstuvwxyz':#a - 97
        == 97+3=100 = d
        number+=shift
        if character.isupper():#A
            if number>ord('Z'):
                number-=26#Z - 90 >> 90+3=93 >> ] >> 93-26=67= C
            elif number<ord('A'):
                number+=26
        else:
            if number>ord('z'):#z = 122 >> 122+3=125
                number-=26
            elif number<ord('a'):
                number+=26
        print(chr(number))
        secreteMessage=secreteMessage+chr(number)
        print(secreteMessage)
    else:
        secreteMessage=secreteMessage+character
        print(secreteMessage)

```



8.Chapter 15 Activity 18

CODING AN ENCRYPTION ALGORITHM

In the language you are studying, create and test a program to encrypt or decrypt a message using a key entered by a user.

```
#ch15_activity18
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
LETTERS=LETTERS.lower()
def encrypt(message,key):#hello,3
    encrypted=""
    for chars in message:
        if chars in LETTERS:
            num=LETTERS.find(chars)#h=7
            num+=key#7+3=10 ==k
            encrypted+=LETTERS[num]
    return encrypted
def decrypt(message,key):#khoor
    decrypted=""
    for chars in message:
        if chars in LETTERS:
            num=LETTERS.find(chars)#k
            num-=key
            decrypted+=LETTERS[num]
    return decrypted
def main():
    message=input("Enter your message:")
    key=int(input("Enter your key 1-26:"))
    choice=input("Encrypt or Decrypt (e or d):")
    if choice.lower().startswith('e'):
        print(encrypt(message,key))
    else:
        print(decrypt(message,key))
#main
main()
```

```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
===== RESTART: D:/SMSD/HLIS_CS/CS_OL_N/Python/ch15
_activity18.py =====
Enter your message:hello
Enter your key 1-26:3
Encrypt or Decrypt (e or d):e
khoor
>>>
===== RESTART: D:/SMSD/HLIS_CS/CS_OL_N/Python/ch15
_activity18.py =====
Enter your message:khoor
Enter your key 1-26:3
Encrypt or Decrypt (e or d):d
hello
>>>
```



9. Chapter 16 Activity 3

CODING A PASSWORD ALGORITHM

Code the password algorithm from Activity 2 in the high-level programming language you are learning. You could extend it to include a 'medium' output for passwords that meet only two of the criteria.

```
#ch16_act3
def is_strong_password(password):
    if len(password)<8:
        return False
    has_uppercase=False
    has_special_char=False
    special_char="!@#$%^&*()_+{}[]:;<>,.?~"

    for char in password:
        if char.isupper():
            has_uppercase=True
        if char in special_char:
            has_special_char=True

    return has_uppercase and has_special_char
#main
password=input("Enter password:")
if is_strong_password(password):#True
    print("Password is strong:")
else:
    print("Password is weak:")
```

10. Chapter 17 Activity 16

1 Simulate a traffic light control system by writing a computer program. It must cycle through the normal sequence of lights, printing out the current state. Your tutor might give you the opportunity to use your program to control a real light output.

```
#ch17_activity16
import time
from time import sleep
start='n'
while start!='y':
    start=input("Press y to start")
    for sequence in range(16):
        light=['red','red and amber','green','amber'][sequence%4]
        print(light)
        time.sleep(1.5)
```



2 Draw a flowchart showing the processing required by a chip-and- PIN credit card reader. This is used when you pay for something in a shop. It collects data about the purchases from the till, gets your card details, checks your PIN, sends encrypted data to the bank and waits for a reply. It might print a receipt if the purchase is allowed and informs the till of the result.

```
SET attempts TO 0

WHILE attempt<>3 THEN
    SEND 'Enter PIN' TO DISPLAY
    RECEIVE pin FROM KEYBOARD
    IF pin==111 THEN
        SEND 'Approve Payment' TO DISPLAY
    ELSE
        SET attempt TO attempt+1
ELSE
    SEND 'Lock Card' TO DISPLAY
```

```
attempts = 0

while attempts < 3:
    # Prompt the user to enter the PIN

    pin = int(input("Enter PIN:"))

    # Check if the entered PIN is correct
    if pin == 111:
        print("Approve Payment")
        break # Exit the loop if the PIN is correct
    else:
        attempts += 1
        print(f"Incorrect PIN. {3 - attempts} attempts
remaining.")

# Check if the maximum attempts are reached
if attempts == 3:
    print("Lock Card")
```

```
>>> %Run ch17_a16.py
Enter PIN:
112
Incorrect PIN. 2 attempts remaining.
Enter PIN:
113
Incorrect PIN. 1 attempts remaining.
Enter PIN:
111
Approve Payment
```



11.Chapter18 Activity7

The game is over either when the player's score exceeds 1 million or if the player runs out of 'health' and isn't operating in 'god mode'.

```
IF year=11 AND (grad<target OR target>7) THEN
    SET revision_class TO True
```

```
END IF
```

```
yearGroup=int(input("Enter year group"))
grade=int(input("Enter grade (9-1)"))
target=int(input("Enter target grade (9-1)"))

if yearGroup==11 and grade<target:
    print("You should attend the revision class")
else:
    print("No need to attend the revision class.")
```

12.Chapter18 Activity18

Mohammad has decided to expand his gardening business to give online advice. He is writing a web page that advises customers which fertiliser to buy from his online shop. The web page will ask customers about their plants and then give relevant advice using these rules:

■If plants have yellow leaves mainly near the soil, then they need a nitrogen fertiliser.

■If plants have brown leaves that are small, then they need a phosphorous fertiliser.

To program this, you first need to work out what are the important features in the rules - these will be your variables. In this case, the colour of the leaves is important. We can write the rules in pseudocode like this.

```
SET advice TO 'nothing'
```

```
IF colour='yellow' THEN
```

```
    IF leaftips=TRUE THEN
```

```
        SET advice TO 'magnesium'
```

```
    ELSE
```

```
        SET advice TO 'nitrogen'
```

```
    END IF
```

```
ELSE
```

```
    IF colour='brown' AND size='small' THEN
```

```
        SET advice TO 'phosphorous'
```

```
    ELSE
```

```
        IF colour='brown' AND size='normal' THEN
```

```
            SET advice TO 'potassium'
```

```
        END IF
```

```
ELSE
```

```
    IF leaves='cracked' OR leaves='misshapen' THEN
```

```
        SET advice TO 'calcium'
```

```
    END IF
```

```
END IF
```



END IF
SEND advice TO DISPLAY

```
colour=input("Enter colour: ")
size =input("Enter size: ")
leaves=input("Enter leaves condition (cracked/misshapen/other):")
leaftips=input("Enter leave tips (True or False)")

advice = 'nothing' # Default advice

if colour == 'yellow':
    if leaftips == 'True':
        advice = 'magnesium'
    else:
        advice = 'nitrogen'
elif colour == 'brown':
    if size == 'small':
        advice = 'phosphorous'
    elif size == 'normal':
        advice = 'potassium'
else:
    if leaves == 'cracked' or leaves == 'misshapen':
        advice = 'calcium'

print("Advice:", advice)
```



13.Chapter18 Activity19

CHANGING CONDITIONS

- 1 Why would the code for Nadia not work if the last two lines were swapped around?
- 2 How can you change the conditions so that it works with the code either way round?

```

SET alarm TO '7:30 am'
IF term_time = false OR (day = 'Saturday' OR day = 'Sunday')
THEN
    SET alarm TO '9:00 am'
ELSE
    IF term_time = true AND day = 'Saturday' THEN
        SET alarm TO '8:00 am'
    END IF
END IF

```

```

alarm = '7:30'
term_time = input('Term time?')
day = input('Day?')
if term_time == 'y' and day == 'Saturday':
    alarm = '8:00'
else:
    if term_time == 'n' or (day == 'Saturday' or day == 'Sunday'):
        alarm = '9:00'
print(alarm)

```

```

Term time? (y or n)y
DayMonday
7:30

```

```

===== RESTART: D
Term time? (y or n)y
DaySaturday
9:00

```



14. Chapter 19 Activity 20

UNDERSTANDING LOGIC STATEMENTS

1 Work out what happens for the following statement.

```
IF gender = 'female' AND (subject = 'computer science'
OR subject = 'physics') THEN
    SEND "Superstar!" TO DISPLAY
END IF
```

- a for a girl studying French
- b for a girl studying computer science
- c for a boy studying physics.

2 Which answer would be different if the programmer had forgotten the brackets?

```
gender=input("Enter gender (male or female):")
subject=input("Enter subject (cs or physics)")
if gender=='female' and (subject=='cs' or subject=='physics'):
    print("Superstar:")
```

15. Chapter 18 Checkpoint S2

Produce a logic statement for the operation of headlamps on a car - the headlamps come on if the light sensor reading is dark or if the headlamp switch is turned on.

```
# Assuming sensor_reading and headlamp_switch are variables
representing the sensor reading and headlamp switch status

sensor_reading = input("Enter sensor reading (dark or light):")
sensor_reading = sensor_reading.lower()
headlamp_switch = input("Is the headlamp switch on? (yes or no):")
headlamp_switch = headlamp_switch.lower()

# Check conditions and send signal to turn on headlamps
if sensor_reading == "dark" or headlamp_switch == "on":
    print("Sending switch-on signal to headlamps")
else:
    print("No signal")
```



16.Chapter 19 Activity 23

ACTIVITY 23**CREATING A SIMULATION**

- 1** Using a programming language that you are learning:
 - a** create a simulation of rolling a die. The die must come up with 1 to 6 with equal probability.
 - b** adapt your program so that the die is biased to give 6 more often than it should.
- 2** In a group, research techniques used in machine learning and illustrate your findings using presentation software. Bring your talks together and present them back to the class.

```
import random
first_roll=random.randint(1,6)
print(first_roll)
```

```
import random
first_roll=random.randint(1,7)

if first_roll>=6:
    print(6)
    print(first_roll,6)
else:
    print(first_roll)
```



17.Chapter 19 C2

Develop a program to model the populations of rabbits and foxes in a habitat. Use the following information to help you.

- Foxes eat rabbits.
- Foxes die of starvation if they don't eat enough rabbits.
- Rabbits and foxes both reproduce, but rabbits reproduce faster.
- Rabbits and foxes both die of other causes.

```
# Initialize rabbit and fox populations
initial_rabbit_population = 50
initial_fox_population = 10

# Define model parameters
growth_rate = 0.05
interaction_coefficient = 0.0002

# Simulate population changes over 10 iterations
for iteration in range(10):
    # Calculate change in rabbit population
    change_rabbit = growth_rate * initial_rabbit_population -
interaction_coefficient * initial_fox_population *
initial_rabbit_population
    #change_rabbit = growth_rate * initial_rabbit_population

    # Update rabbit population based on the calculated change
    new_rabbit_population = initial_rabbit_population +
change_rabbit

    # Display results for each iteration
    print(f"Iteration {iteration + 1}: Rabbit Population =
{new_rabbit_population:.2f}")

    # Update initial rabbit population for the next iteration
    initial_rabbit_population = new_rabbit_population

# Output the final rabbit population
print(f"Final Rabbit Population: {new_rabbit_population:.2f}")
```



18.Ch20 Unit Question 8

Yusuf is developing a logic circuit for an automatic door system. The output from the sensor on the street (S) is 1 when someone approaches the door from outside.

The output from the sensor in the building (B) is 1 when someone approaches the door from inside.

The opening mechanism works only if the manual switch (M) is off (0) Develop a suitable IF statement in pseudocode to represent this logic.

```
IF (S=1 OR B=1) AND M=0 THEN  
    SEND open To door_opening_mechanism  
END IF
```

```
S = int(input("Enter street sensor (0 or 1):"))  
B = 0 # Replace with the actual value of B  
M = 0 # Replace with the actual value of M  
  
if (S == 1 or B == 1) and M == 0:  
    command_to_send = 'open'  
    print(f"Sending {command_to_send} to door_opening_mechanism")  
else:  
    print("Nothing to do")
```



19.Ch20 Unit Question 9

9 Develop a pseudocode representing the logic for a cinema ticket pricing application.

- Standard pricing is £5
- Children under 4 are free.
- Children under 16 and people over 65 are half price.

SET standardPrice TO 5

IF age<4 THEN

 SET price TO 0

ELSE

 IF age<16 OR age>65 THEN

 SET price TO standardPrice/2

 ELSE

 SET price To standardPrice

 END IF

END IF

```
standard_price = 5 # Replace with the actual standard price
#age = 10 # Replace with the actual age value
age = int(input("Enter your age:"))
if age < 4:
    price = 0
else:
    if age < 16 or age > 65:
        price = standard_price / 2
    else:
        price = standard_price

# You can then use the 'price' variable as needed in the rest of
your code
print(f"The calculated price is: {price}")
```



20.Chapter 22 Activity 7

RESTRICTING ACCESS

Think about how you would write a program so that only approved users who enter their password correctly can get beyond the login screen.

- 1 Develop the pseudocode for the username and password checking part of the program. Then, write your program.
- 2 How secure do you think your implementation is? Could it be improved?

```
#ch22_activity7
def read_in_data():
    rawData=open('users.txt','r')
    inputData=rawData.readlines()
    rawData.close()
    users=[]
    index=0
    for line in inputData:
        users.append(inputData[index].split(','))
        index+=1
        for names in users:
            names[1]=names[1].rstrip()
    return users

def check_user_name():
    attempt=1
    nameCorrect=False
    while attempt<4 and nameCorrect==False:
        print('\n Username attempt:',attempt)
        nameEntered=input("Enter your username:")
        valid=False
        index=0
        while valid==False and index<length:
            if users[index][0]==nameEntered:
                valid=True
            else:
                index+=1
        print(users[index][0])
        if valid==False:
            attempt+=1
        else:
            nameCorrect=True
        return index

def enter_password(position):
    attempt=1
    password=''
    while attempt<4 and password!=users[position][1]:
```



```
        print('\n Password attempt:',attempt)
        password=input("Enter your password:")
        if password==users[position][1]:
            print('Correct Password entered.')
        else:
            print('\nPassword incorrect')

#main program
users=read_in_data()
length=len(users)
positionlist=check_user_name()
print(positionlist)
if positionlist<length:
    enter_password(positionlist)
else:
    print('\nContact the system administrator')
```

