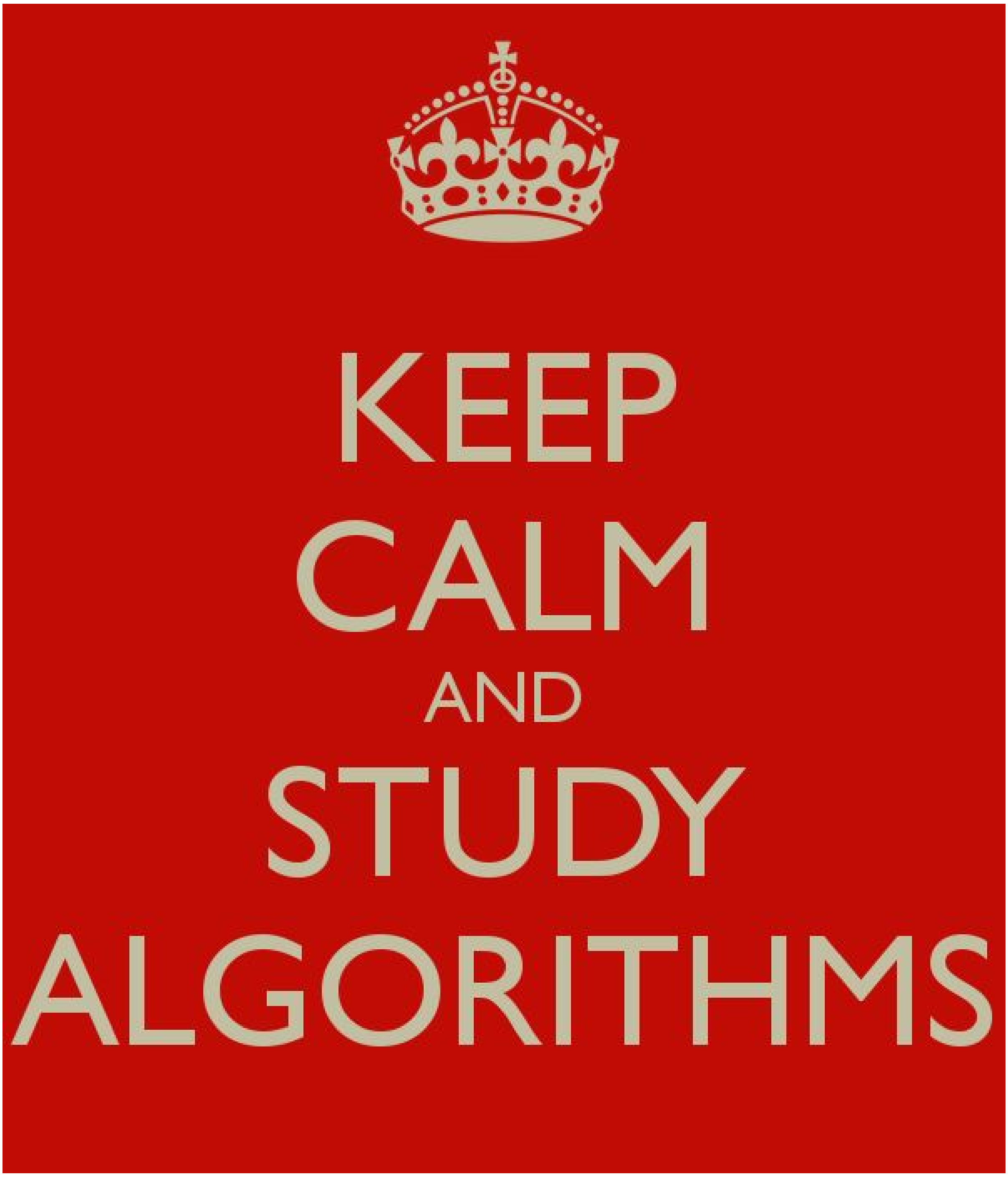
NAME: Manraj Lally



Contents

Tasks

Algorithms

In chess how many possible moves are there when each player has made three moves?

After three moves there are 9 million possible moves available.

And when each player has made four moves?

After 4 moves there are over 288 billion possible moves

State another scenario where very complex algorithms are used.

Algorithms are used for Internet searches to make them quicker and the results more relevant to the user. They will even auto-complete the search terms based on previous results. Algorithms are also used in auto-pilot systems in airplanes.

Describe an algorithm.

A set of instructions that are used to complete tasks.

Below explain the steps of riding a bike. You might prefer to show this on hand drawn paper, take a photo and insert below.

Go outside
Get bike
Sit on bike seat
Place left hand on left handlebar and place right hand on right handlebar
Place left foot on left pedal or right foot on right pedal.
Push off from the ground with the foot that isn’t on the pedal.
Push down your foot on (left or right) the pedal
Place foot that pushed off from ground on the pedal.
After pushing down one foot, push down the other one
Repeat the step above
Stop at destination by pressing brakes on your handlebars with your hands.

Sequence, Iteration and Selection

Explain which parts of the algorithm above are SEQUENCE, ITERATION and SELECTION.

In the algorithm above, ‘repeat’ is iteration because it is iterating. The selection is deciding what to do. This is seen in the line ‘Place left foot on left pedal or right foot on right pedal’ because there is a decision and also ‘stop at destination’ because you choose your destination. The other commands are sequences.

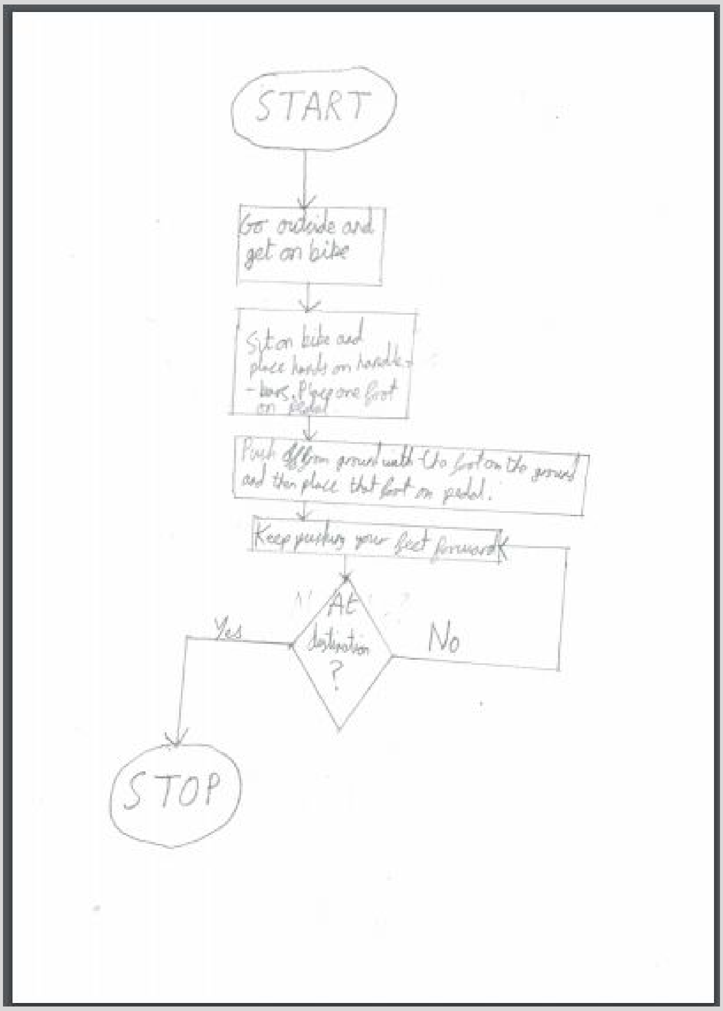
What are the two most important factors that make algorithms successful?

1) Correctness
2) Efficiency

Flowcharts

Instead of just writing a bullet pointed list we can use a diagram to explain an algorithm. These diagrams are called FLOWCHARTS.

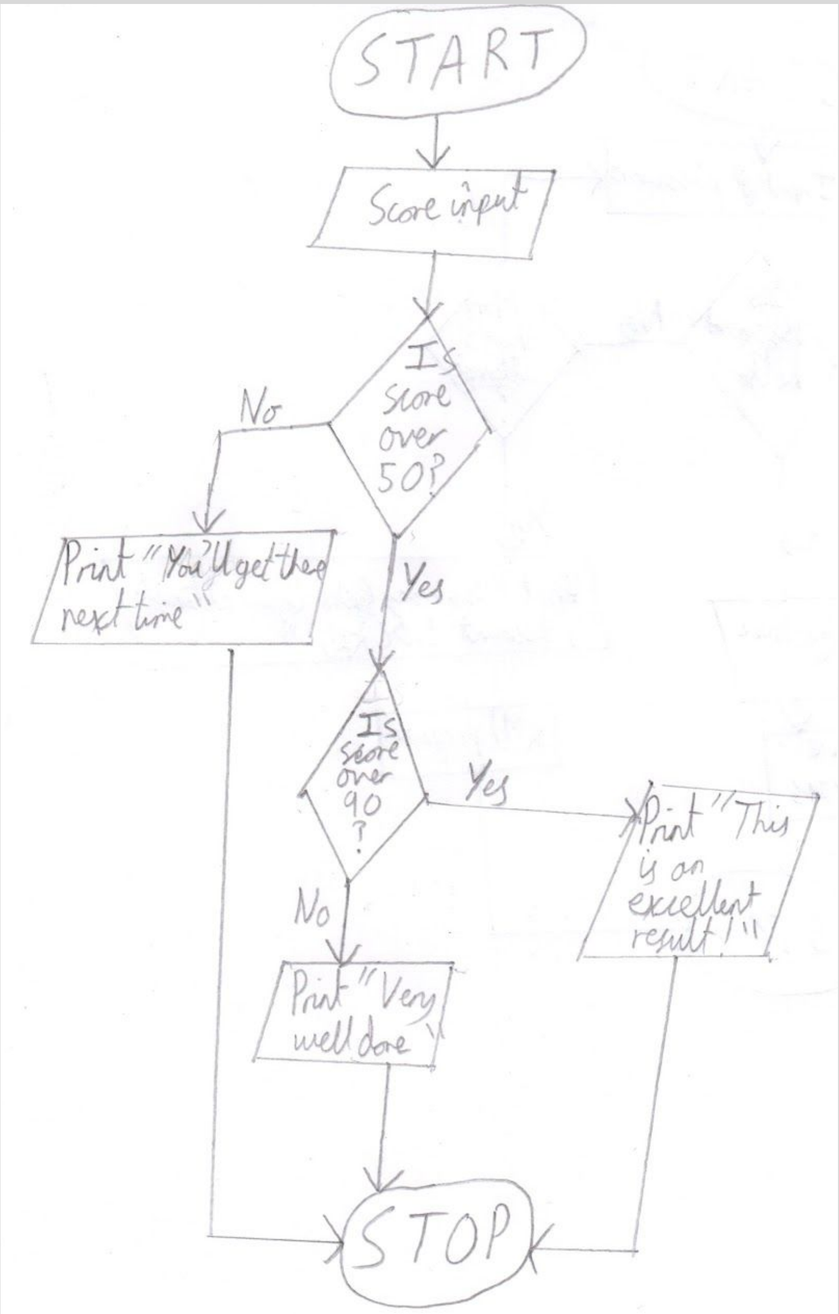
Using a FLOWCHART to show the algorithm above redo the process of riding a bike. Annotate where you are using SEQUENCE, ITERATION and SELECTION. This may be quicker and easier done on paper, take a photo and insert below.


START
Go outside and get on bike
Sit on bike and place hands on handle-bars. Place one foot on pedal
Push off from ground with the foot on the ground and then place that foot on pedal.
Keep pushing your feet forward
At destination ?
Yes
No
STOP

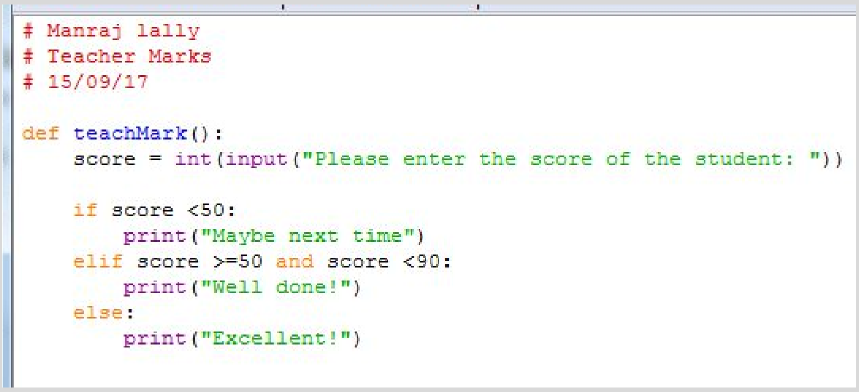
You can see that the diamond box

is a selection and when the selection is no there is iteration. The rest of it is sequence.

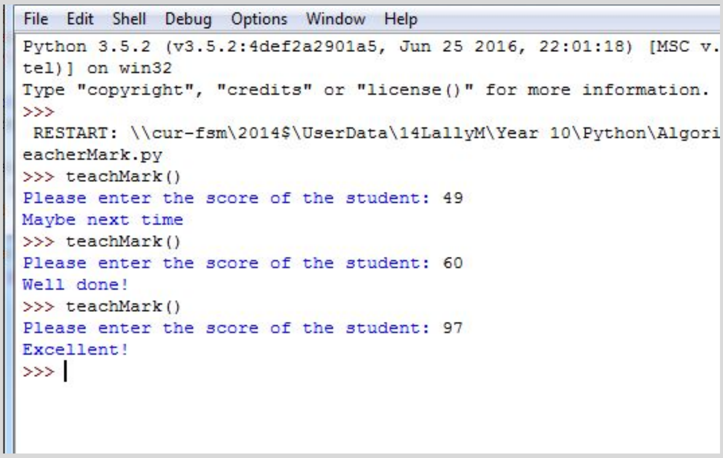
Use a flowchart create an algorithm to solve this problem. A teacher is marking his students test papers. If they achieve over 50% he would like the message ‘Very well done’ displayed. If they achieve over 90%, they should also receive a second message stating ‘This is an excellent result’. If they score 50% or lower, the message will be ‘You will get there next time’. It might be quicker and easier to do this on paper, take a photo and insert below.



Code the above in Python. Screen shot your code and insert it below. Annotate using comments where there is SEQUENCE, SELECTION and ITERATION.



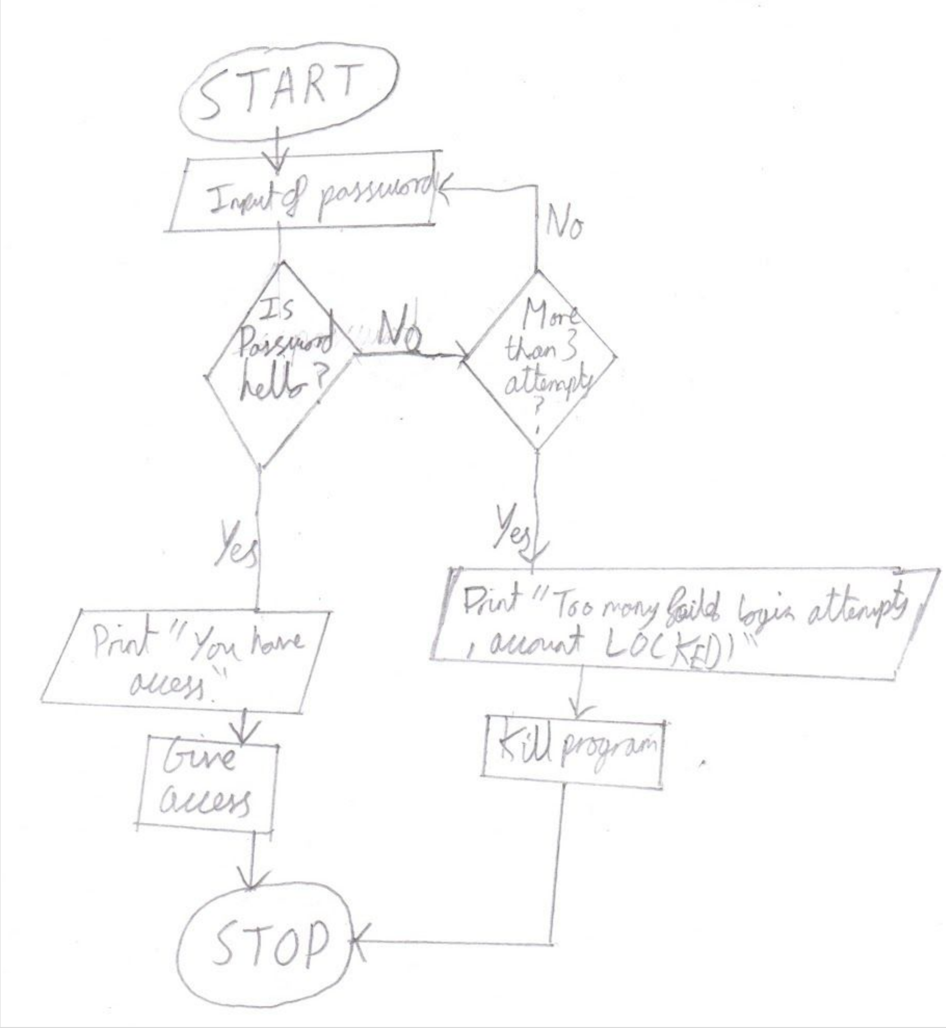
```
# Manraj lally
# Teacher Marks
# 15/09/17

def teachMark():
    score = int(input("Please enter the score of the student: "))

    if score <50:
        print("Maybe next time")
    elif score >=50 and score <90:
        print("Well done!")
    else:
        print("Excellent!")
```



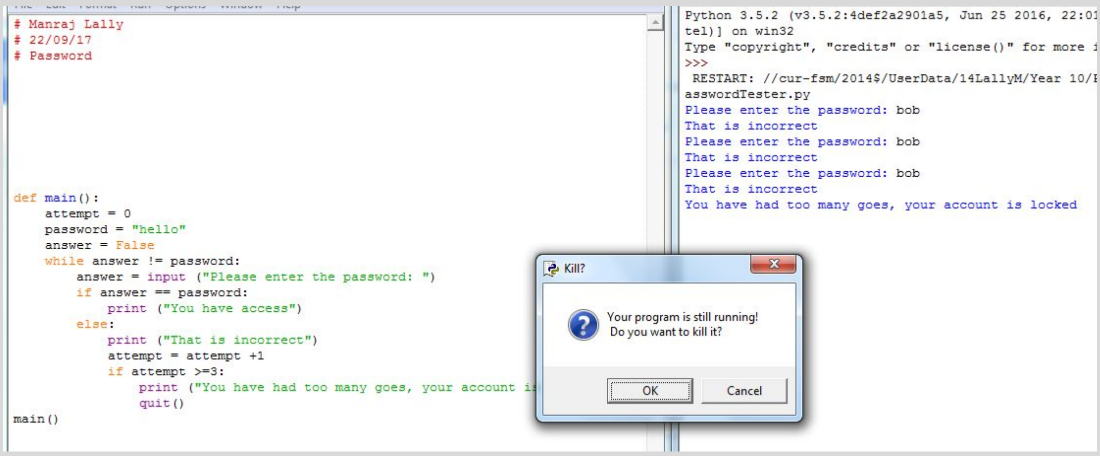
```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: \\cur-fsm\2014$\UserData\14LallyM\Year 10\Python\Algori
eacherMark.py
>>> teachMark()
Please enter the score of the student: 49
Maybe next time
>>> teachMark()
Please enter the score of the student: 60
Well done!
>>> teachMark()
Please enter the score of the student: 97
Excellent!
>>> |
```

It's all sequence apart from when the program has to decide if the score is below 50, above 50 or above or below 90. There is no iteration.

Use a flowchart create an algorithm to solve this problem. Asking a user for a password. Allowing the user to have three attempts at the password before "Too many failed login attempts, account LOCKED!" is displayed. It might be quicker and easier to do this on paper, take a photo and insert below.



Code the above in Python. Screen shot your code and insert it below. Comment where there is SEQUENCE, SELECTION and ITERATION.

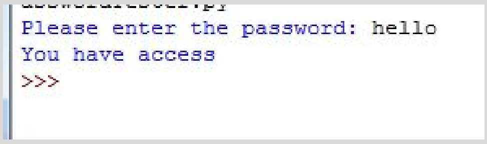


```
# Manraj Lally
# 22/09/17
# Password







def main():
    attempt = 0
    password = "hello"
    answer = False
    while answer != password:
        answer = input ("Please enter the password: ")
        if answer == password:
            print ("You have access")
        else:
            print ("That is incorrect")
            attempt = attempt +1
            if attempt >=3:
                print ("You have had too many goes, your account i
                quit()
main()
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01
tel)] on win32
Type "copyright", "credits" or "license()" for more i
>>>
 RESTART: //cur-fsm/2014$/UserData/14LallyM/Year 10/P
asswordTester.py
Please enter the password: bob
That is incorrect
Please enter the password: bob
That is incorrect
Please enter the password: bob
That is incorrect
You have had too many goes, your account is locked
```

Kill?

Your program is still running!
Do you want to kill it?



```
Please enter the password: hello
You have access
>>>
```

It is all sequence apart from when the password is incorrect and the program repeats the question which is iteration. Deciding if the password is correct or not is selection

Variables and Constants

Define the term VARIABLE and CONSTANT and explain how they are different.

A variable is a value that may change whilst a program is running. A constant is a value that will stay the same whilst running.

Describe what naming conventions you should use when creating variables.

When naming variables, there are two methods you can use to help you. The first is camel case which means that when a new word is written, a capital letter is used. For example salesOfIcecreams. You can also use underscores after every word which looks like sales_of_icecreams.

Pseudocode

Describe pseudocode stating its purpose and advantages.

Pseudocode is a kind of structured English for describing algorithms. It’s a generic, code-like language that can be easily translated into any programming language. Writing in pseudocode helps you concentrate on the logic(the process) and efficiency of your algorithm before you have to start thinking about the code you will use. It’s important to check that you have included all the stages in your process as it is easier to spot anything missing at this stage than when you have carefully translated it into code!

Operators

Describe the terms OPERATOR and OPERANDS. If it helps draw a diagram. This can be done on paper, take a photo and insert below.

An operator is a symbol that tells the computer to perform a specific action on the data and manipulate it in a particular way. The data on which an operator performs the action is called an operand.

Complete the table below

Operator	**Purpose**	**Example**

+	Addition: It adds a value to another value.	5 + 5 = 10
-	Subtraction: It subtracts a value from another value.	10 - 5 = 5
*	Multiplication: It multiplies a value with another value.	5 * 5 = 25
/	Division: It divides a value by another value.	13 / 3 = 4.333 totalSweets/totalChildren
DIV	Quotient: Like division but it only returns the whole number or integer.	13 DIV 3 = 4 totalSweets\totalChildren
MOD	Modulus: This will return the remainder of the division	13 / 3 = 4 remainder 1. Therefore: 13 MOD 3 = 1
^	Exponentiation: This is for powers of.	3^3 = 27. It is the same as writing 3^3
==	Equal to: Checks if two values are equal.	length == width
=	Assignment: Can be used for variables and constants to assign them a value.	totalMade = costOfIcecream*sales
!=	Not equal to: Checks if the two values are not equal.	temperature != 20
<	Less than: Checks to see if one value is less than the other.	temperature < 20
>	Greater than: Checks to see if one value is greater than the other.	temperature > 20
<=	Less than or equal to: Checks to see if one value is less than or equal to another.	temperature <= 20
>=	Greater than or equal to: Checks to see if one value is greater than or equal to another.	temperature >= 20

Describe the difference between using = and ==?

= is the symbol for assignment which is usually used when giving a variable or constant a value. == is the symbol to check if two values are equal to each other.

Provide examples from the list above of mathematical / **arithmetic** operators.

+, -, *, /, DIV, MOD, ^

Provide examples from the list above of comparison / **relational** operators.

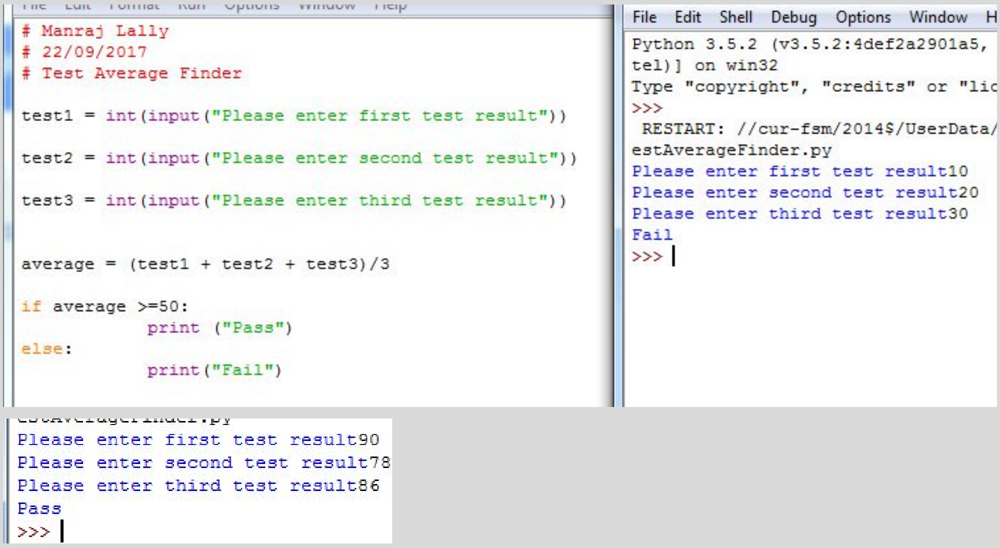
==, =, !=, <, >, <=, >=

IF, ELSE ELSE IF

A teacher would like a program that allows her to enter three test results and calculate the average. Then if the average is 50 or above it should output the message ‘PASS’ and if it is below 50 the message should be ‘FAIL’. Design using pseudocode the algorithm for this problem.

```
test1 = input(“Please enter first test result.”)
test2 = input(“Please enter second test result.”)
test3 = input(“Please enter third test result.”)

Average = (test1 + test2 + test3)/3

If average >= 50 then
            Print (“Pass”)
Else
            print(“Fail”)

endif
```

Code the above. Screenshot your code and insert the image below.



```
# Manraj Lally
# 22/09/2017
# Test Average Finder

test1 = int(input("Please enter first test result"))

test2 = int(input("Please enter second test result"))

test3 = int(input("Please enter third test result"))


average = (test1 + test2 + test3)/3

if average >=50:
            print ("Pass")
else:
            print("Fail")
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5,
tel)] on win32
Type "copyright", "credits" or "lic
>>>
 RESTART: //cur-fsm/2014$/UserData/
estAverageFinder.py
Please enter first test result10
Please enter second test result20
Please enter third test result30
Fail
>>> 
```

```
Please enter first test result90
Please enter second test result78
Please enter third test result86
Pass
>>> 
```

Describe the difference between ELSE and ELSEIF.

Elseif is used in the middle of an if statement, else only (and always) is used at the end.

Describe the purpose of indentation

Indentation makes code easier to read as well as easier to understand.

Boolean Operators

Complete the table below (Example Code)

Operator	Purpose	Example
AND	Logical AND operator If all the operands are true then the condition becomes true.	If length > 6 AND width > 3 then Area = length * width Else Print ("Rectangle is not large enough.") Endif Note: In this example the length must be greater than 6 AND the width must be greater than 3 to work out the area.

OR	Logical OR operator
	If any of the operands are true then the condition becomes true
		score = int(input("Please enter score: "))
		if score < 0 or score >100:
		        print("This cannot be right please check")

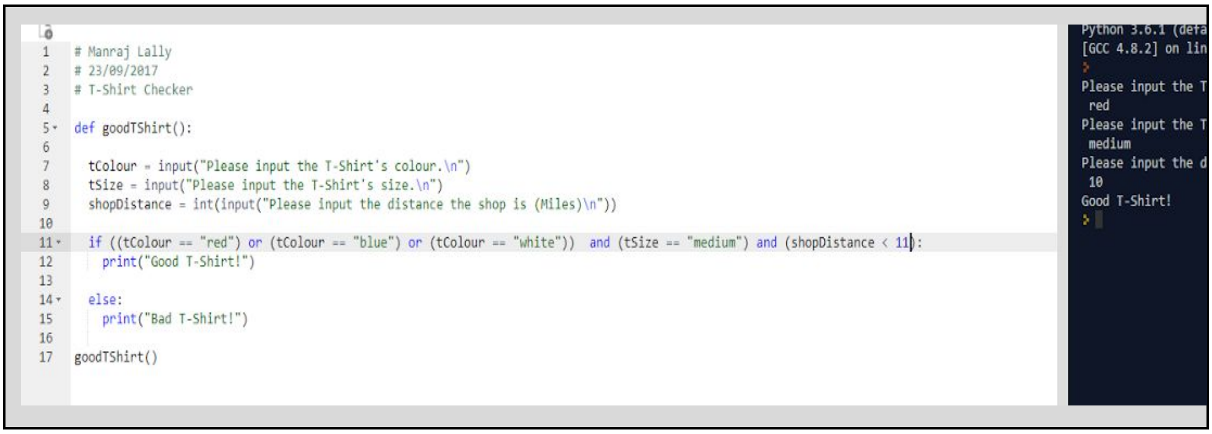
NOT	Logical NOT operator
	Used to reverse the logical state of the operand
		length = int(input("Please enter length: "))
		    width = int(input("Please enter width: "))
		    if not (length >=6 and width >=3):
		        print("Not big enough")

A student would like to select a suitable T-shirt from local shops. The colour could be red, blue or white, the size should be medium and the shop must be no more than 10 miles away. Design using pseudocode the answer to this problem.

```
Colour = input ("Enter colour of T-shirt.")

Size = input ("Enter size as S, M or L.")

Distance = input (" Enter distance to shop in miles.")
```

```
if(colour == “red” or colour == “blue” or colour == “white”) and (size == “medium”) and
(Distance < 11)
        Print (“Good T-shirt”)

Else:
        Print (“Bad T-shirt”)
```

Code your design. Screenshot and inset the image below.



```python
# Manraj Lally
# 23/09/2017
# T-Shirt Checker

def goodTShirt():

  tColour = input("Please input the T-Shirt's colour.\n")
  tSize = input("Please input the T-Shirt's size.\n")
  shopDistance = int(input("Please input the distance the shop is (Miles)\n"))

  if ((tColour == "red") or (tColour == "blue") or (tColour == "white"))  and (tSize == "medium") and (shopDistance < 11):
    print("Good T-Shirt!")

  else:
    print("Bad T-Shirt!")

goodTShirt()
```

```
Python 3.6.1 (defa
[GCC 4.8.2] on lin
>
Please input the T
 red
Please input the T
 medium
Please input the d
 10
Good T-Shirt!
>
```

Nested IF Statements

What does the term ‘Nested If Statements’ mean?

The ‘if … elseif … else’ statements are sometimes referred to as nested if statements as the ‘elseif’ and ‘else’ statements run inside the overall ‘if’ statement. But strictly speaking, ‘nested if’ should refer to two complete ‘if’ statements, one running inside the other. Basically an ‘if’ statement inside an ‘if’ statement.

Switch / CASE

Show using pseudo code how a solution using multiple IF Statements can also be completed using Switch/CASE. Use the same context for both the IF Statement and Switch/CASE.

```
Pseudo code for IF STATEMENTS
def UsingIF():
        answer = input ("Please enter an option.")
        if answer == "A":
                print ("Sorry that is not right.")
        elif answer == "B":
                print ("Sorry that is not right.")
        elif answer == "C":
                print ("That is right.")
        elif answer == "D":
```

```
        print ("Sorry that is not right.")
    else:
        print("You have not made a valid choice")
UsingIF()
```

```
Pseudo code for Switch/CASE
def UsingSwitchCase():
    answer = input ("Please select an option")

    switch answer:
        case "A":
            print("Sorry.  That is not right")
        case "B":
            print("Sorry.  That is not right")
        case "C":
            print("That is right")
        case "D":
            print("Sorry.  That is not right")
        default:
            ("You have not made valid choice")
```

Key Terms

Key Term	Description
Sequence	Sequence is a basic step such as ‘sit down’ that requires no decision to be made and doesn’t loop without an iteration.
Sub Tasks	Sub Tasks are a task that is part of a larger task.
Decision	Decision is when there are two or more ‘paths’ that can be taken and choosing which one is decision. Like selection.
Process	A process is when a computer program is being executed.
Variable	A variable is a value that may change whilst a program is running.
Pseudocode	Pseudocode is a kind of structured English for describing algorithms. It’s a generic, code-like language that can be easily translated into any programming language.
Identifier	This is a name for something or what it is identified as. Print is an identifier because it’s the name of the function that is print.
Constant	A constant is a value that will stay the same whilst running.

Comment	A comment is usually what a programmer will add to their code to help them understand what section does what. They can be written like //Something….
Operator	An operator is a symbol that tells the computer to perform a specific action on the data and manipulate it in a particular way.
Operand	The data on which an operator performs the action is called an operand.
Parentheses	These are pretty much brackets. Usually a word or a phrase is inserted into them help explain something.
Relational Operator	A relational operator is an operator that tests or defines some kind of relation between two entities.
Logical Operator	They allow a program to make a decision based on multiple conditions. Each operand is considered a condition that can be evaluated to a true or false value.
Mathematical Operator	These operators are all mathematical. For example + will add a value to another value.

Questions

1. Using pseudocode show how a value of Brian would be assigned to a variable named firstName.

```
firstName = Brian
```

2. Using pseudocode show how a value of 15 and be compared to see if it is equal to 14.

```
15 == 14
```

3. Using pseudocode show how a boolean operators can be used to select all the people in a class that are male with blue eyes but do not have brown hair.

```
if person == male and person == blueEye and person != brownHair
```