NAME: Yousif Baker

Contents

Contents
Tasks
- Algorithms
- Sequence, Iteration and Selection
- Flowcharts
- Variables and Constants
- Pseudocode
- Operators
- IF, ELSE ELSE IF
- Boolean Operators
- Nested IF Statements
- Switch / CASE

Key Terms
Questions

Tasks

Algorithms

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

There are 9 million possible moves

And when each player has made four moves?

After 4 moves there are 288 billion possible moves

State another scenario where very complex algorithms are used.

Algorithms can be used in stock markets to trade shares , they are often used by search engines to speed up the speed in which a result is found.

Describe an algorithm.

An algorithm is a set of instructions that a computer would follow.

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

Place the leg over the bike
Sit on saddle
Place a foot on a peddle
Place the other on the ground
Use the foot on the ground to launch
Place both feet on the pedals
Peddle
Repeat until you have reached the destination

Sequence, Iteration and Selection

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

Sequence - it ensures that instructions are in the right order to prevent error (1 , 2 , 3 .4 , 5 , 6)
Iteration - iterations or “loop” allow algorithms to be simplified by stating certain steps to be repeated unless told otherwise eg (5,6,7)
Selection - At some point in an algorithm there may need to be a question because the algorithm has reached a step where one or more options are available. Depending on the answer given, the algorithm will follow certain steps and ignore others. Eg 3

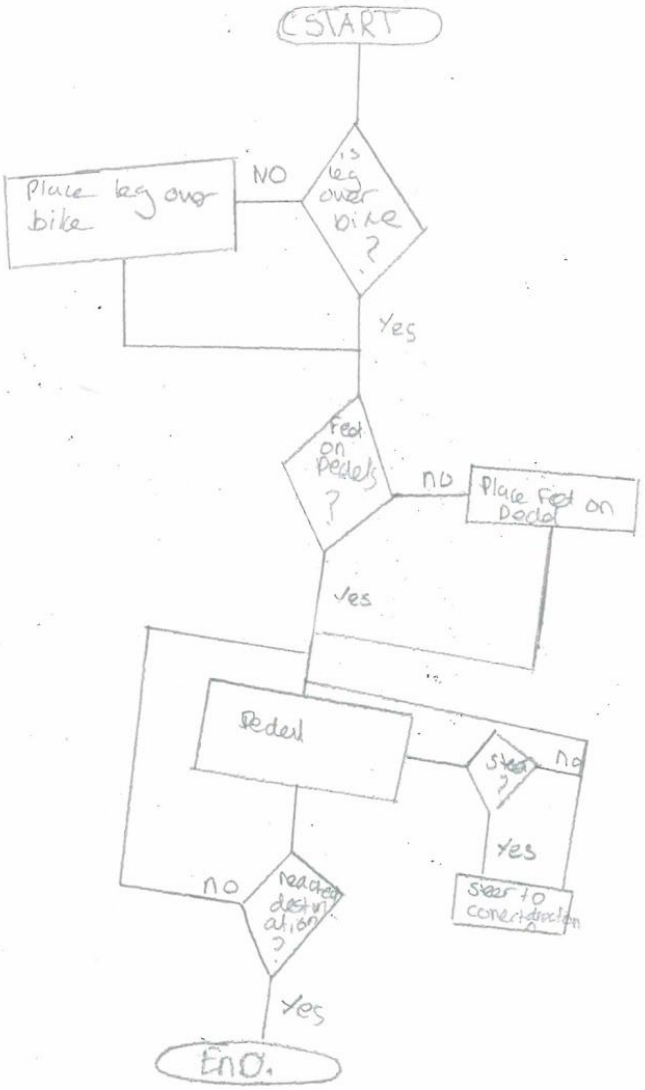
What are the two most important factors that make algorithms successful?

1) correctness (making sure things are right)
2)efficiency (ensuring that everything works efficiently)

Flowcharts

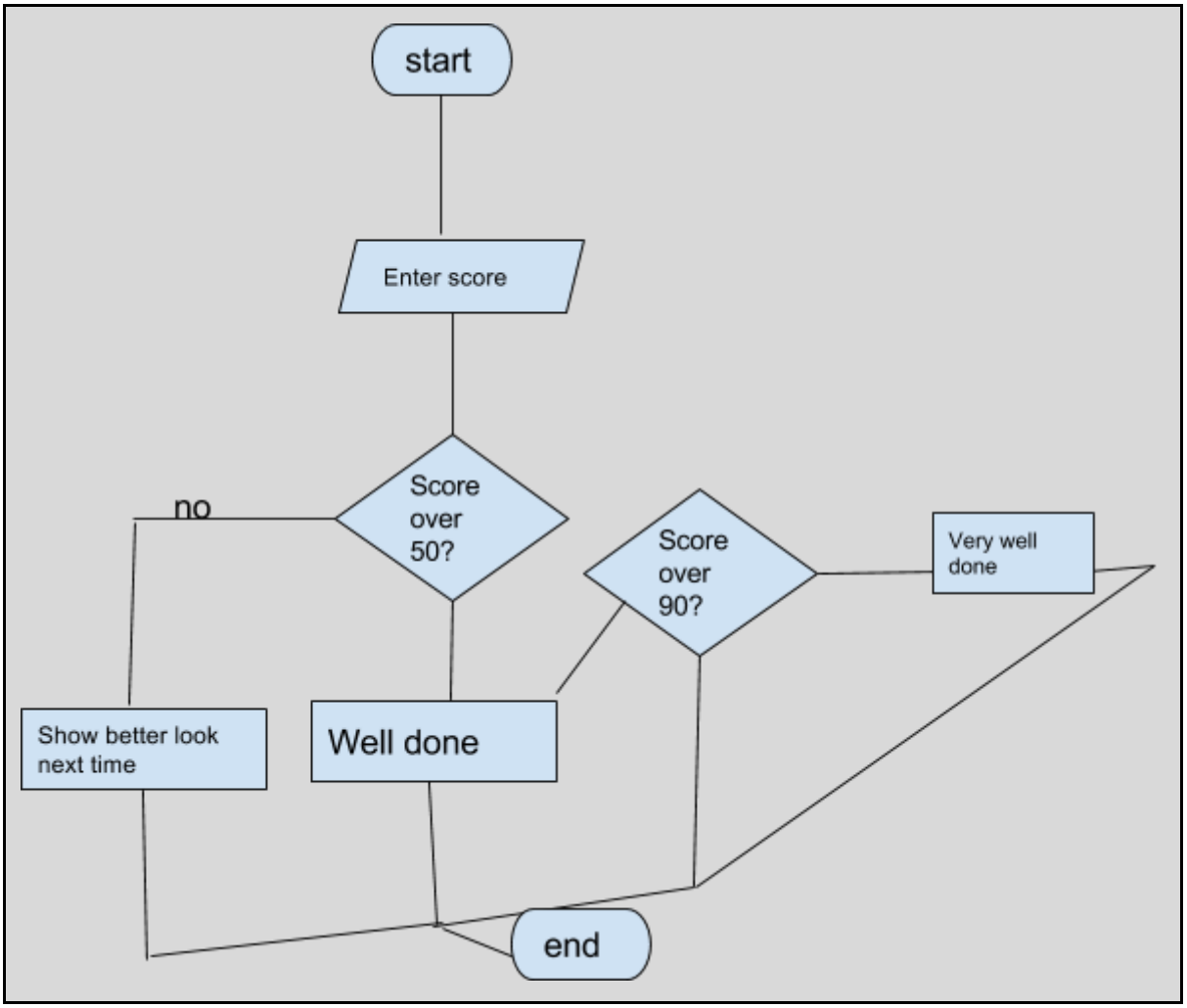
Instead of just writing a bullet pointed list we can use a diagram to explain an algorithm. These diagrams are called FLOWCHARTS.

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
Is leg over bike ?
NO
Place leg over bike
Yes
Feet on Pedals ?
no
Place Feet on Pedal
Yes
Pedal
Steer ?
no
yes
Steer to correct direction
reached destination ?
no
yes
End.

Computing GCSE
Chapter 1: Algorithms WB
6 of 14

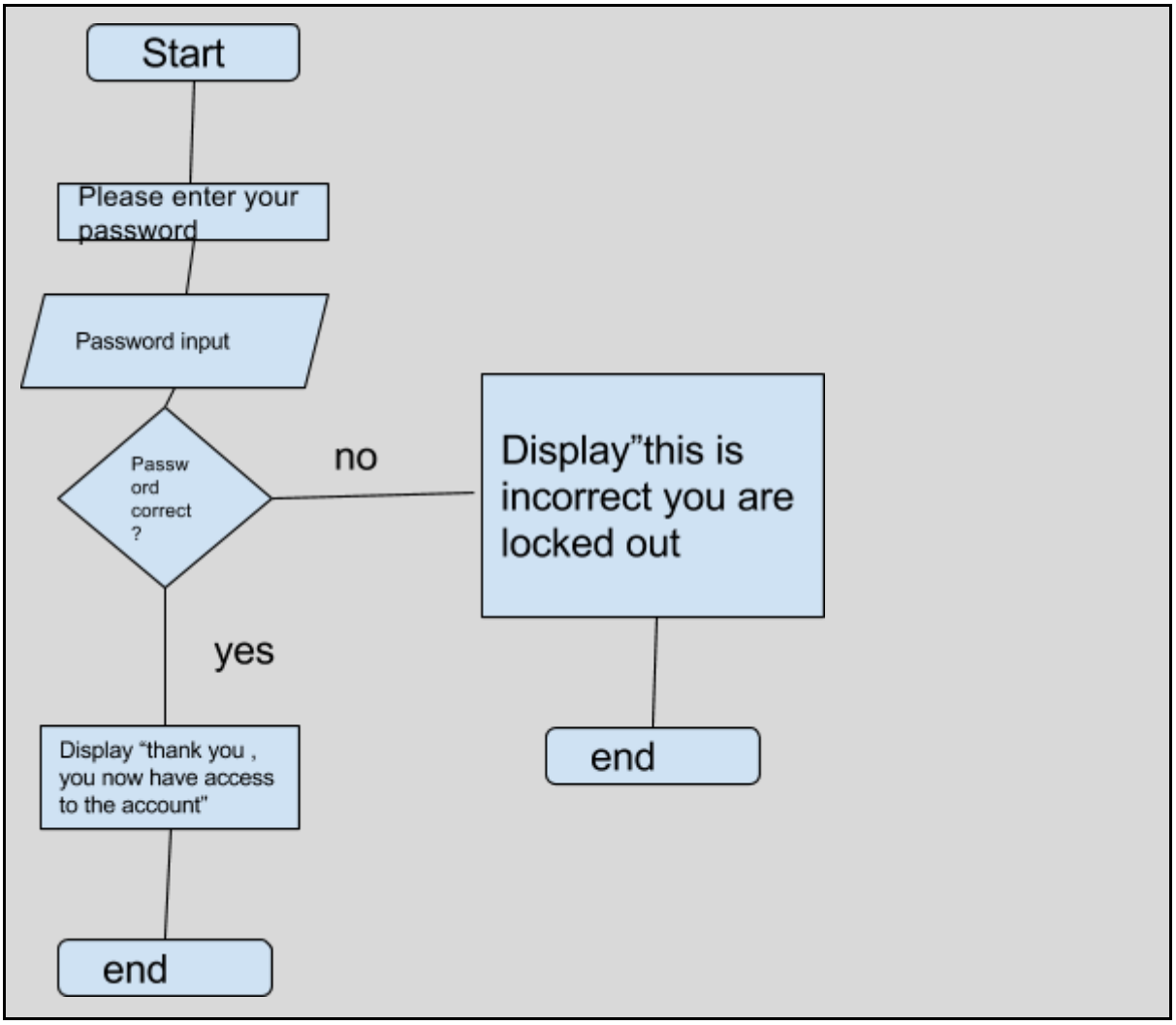
Use a flowchart create an algorithm to solve this problem. A teacher is marking his students test papers. If they achieve over 50% he would like the message 'Very well done' displayed. If they achieve over 90%, they should also receive a second message stating 'This is an excellent result'. If they score 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.

```
score = int(input("What is the test score percentage?"))


if score >=101:
  print ("That is not possible, please check your score")

elif score >=95:
  print ("Very well done this is an excellent result")

elif score <=50:
  print ("You will get there next time")

else:
  print ("Very well done")
```

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.

```
def main():
  attempt = 0
  password = "mySecretPassword"
  answer = False
  while answer != password:
    answer = input ("Enter your password: ")
    if answer == password:
      print ("You have access")
    else:
      print ("Access denied")
      attempt = attempt +1
      if attempt >= 3:
        print ("Incorrect, your account is now locked")
        quit()

main()
```

Variables and Constants

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

Varible = something that can change
A constant is something that will stay the same no matter what.

Describe what naming conventions you should use when creating variables.

Don't use words like 'print' as they already have code attached to them

Pseudocode

Describe pseudocode stating its purpose and advantages.

Its purpose is to allow programmers to know exactly what a person wants so when it comes to programming the program they have to follow a step by step instructions so it allows the developer to be more efficient in their work and allows them to know what they need to do.

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.

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand. For example: >>> 2+3 5. Here, + is the operator that performs addition. 2 and 3 are the operands and 5 is the output of the operation.

Complete the table below

Operator	Purpose	Example
+	Adds two or more variables together	totalCost + tax
-	Subtracts two or more variables from each other	totalCost - offer
*	Multiplication it multiplies two or more variables	Cost * tax
/	Divides two or more variables	pay/time
DIV	use as //) Integer divide	10 DIV 4 = 2 or 10//4 = 2

MOD	Divide two numbers and only print decimal / it only shows the remainders.	3 mod 2 = 5
^	Power of	5^2 = 25
==	If values of two operands are not equal, then condition becomes true.	(a==b) is not true
=	Assigns a value to a variable	Y = 20
!=	It is not equal to	While answer != to username
<	Less than	12<13
>	Greater than	14>12
<=	Less than or equal to	x<=12
>=	more than or equal to	x>=12

Describe the difference between using = and ==?

= assigns a value to a variable and == means if both operands are equal then the variable becomes true

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

+, -, / and *

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

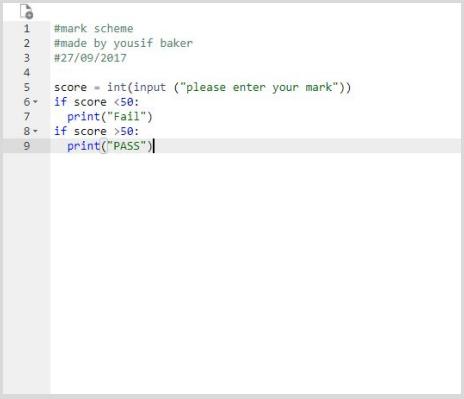
<, >, <= and =>

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.

Print (enter your mark)
If mark <50
Print (FAIL)
If mark >50
Print (pass)

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



```
#mark scheme
#made by yousif baker
#27/09/2017

score = int(input ("please enter your mark"))
if score <50:
  print("Fail")
if score >50:
  print("PASS")
```

Describe the difference between ELSE and ELSEIF.

lfe we had and else it would continuously run the function until the its criteria is met however an elseif

Describe the purpose of indentation

Indentation is important as it makes it easier to read and helps organise the work

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: area = length * width

OR	Logical or operator if any of the operands are true then the condition becomes true	If score < 0 or score >100 then
		print(“this score is invalid”)
		endif

NOT	Logical not operator
	Used to reverse the logical state of the operand.
		If NOT (length >=6 AND width >=3) then
		Print (“rectangle is not large enough.”)
		end if

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.

What colour is shirt?
If answer is not red, blue or white print “Incorrect colour”
What size is shirt?
If answer is not medium print “Incorrect size”
How far away is shop?
If answer > 10 miles print “Incorrect, too far away”

Code your design. Screenshot and inset the image below.

```
def shirt():

  colour= input("please enter a colour of t-shirt , blue, red or
    white: ")
  size = input("please enter size if shirt- s, m or l")
  distence = int(input("please enter distence to the shop :"))

 if (colour == "red" or colour =="blue" or colour =="white") and
   size =="m" and distence <= 10:
    print ("shirt is in stock!")

  else:
    print("sorry , we do not have that shirt in stock")


shirt()
```

Nested IF Statements

What does the term ‘Nested If Statements’ mean?

An if inside another if eg
If purchase > 100 then
    Discount = purchase / 100 * 5
    If discount > 50 then
  Discount = 50

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

Answer = input("Please select an option.")
        If answer == "A" then
        Print ("Sorry, incorrect")

        Elif == "B" then
        Print ("Correct")

        Else then
        Print ("Option not recognised")

        endif
```

```
Pseudo code for Switch/CASE

Answer = input("Please select an option.")
        Case "A":
        Print ("Sorry, incorrect")

        Cse "B"
        Print ("Correct")

  default:
        Print ("Option not recognised")

   endswitch
```

Key Terms

Key Term	Description
Sequence	In Python, sequence is the generic term for an ordered set. There are several types of sequences in Python, the following three are the most important. Lists are the most versatile sequence type. The elements of a list can be any object, and lists are mutable - they can be changed.
Sub Tasks	
Decision	Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome is TRUE or FALSE otherwise. ... Python programming language provides following types of decision making statements
Process	
Variable	Variables are nothing but reserved memory locations to store values and can be functions
Pseudocode	a notation resembling a simplified programming language, used in program design.
Identifier	A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
Constant	a constant is a value that cannot be altered by the program during normal execution
Comment	A sentence that if a hash (#) is used before then the computer will ignore it in the code, used to tell coders what parts is which and to make the code easier to read
Operator	The value that the operator operates on is called the operand. For example: >>> 2+3 5. Here, + is the operator that performs addition. 2 and 3 are the operands and 5 is the output of the operation.

Operand	Operators are special symbols in Python that carry out arithmetic or logical computation. Here, + is the operator that performs addition. 2 and 3 are the operands and 5 is the output of the operation.
Parentheses	
Relational Operator	operators compare the values on either sides of them and decide the relation among them
Logical Operator	The boolean type. A boolean expression (or logical expression) evaluates to one of two states true or false.
Mathematical Operator	An operator is a symbol or function that indicates an operation. For example, in math the plus sign or + is the operator that indicates addition. In Python, we will see some familiar operators that are brought over from math.

Questions

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

```
First name = input “hello , what is your first name?”
```

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

```
14 == 15
```

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 “male” and “blueEyes” not “brownHair”
        Print (“Correct”)
```