

Unit 6 & 7

BSB Year 10

**Cambridge IGCSE™
Computer Science 0478**

Answer Booklet



THE BRITISH
SCHOOL OF
BEIJING, SHUNYI



Mr. Teasdale – The British School of Beijing

This document contains all the answers to the worksheets and homework are supposed to have completed this year for the specific unit listed on the front cover. This will be useful to go through any incorrect answers for revision purposes or to catch up on any work that you missed. The answers are written in **RED**.

Contents

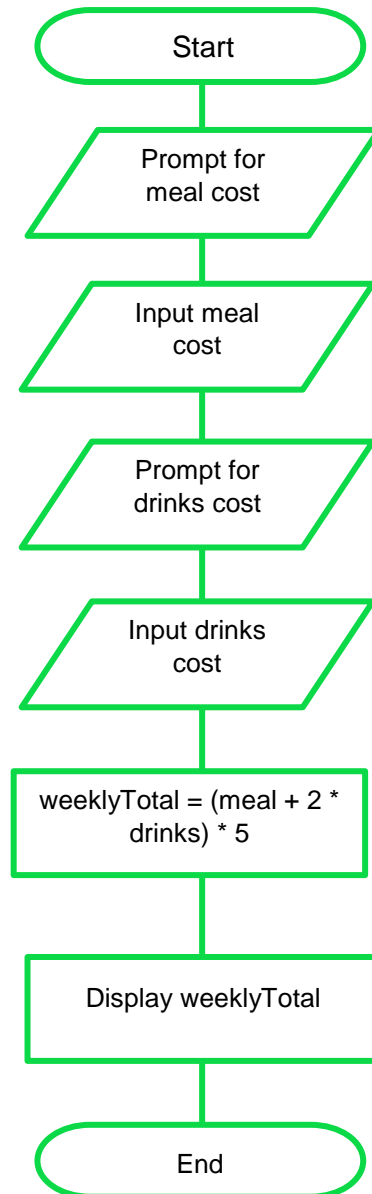
Unit 7 Algorithms and Flowcharts Answers	3
Worksheet 1: Algorithms and flowcharts	3
Homework 1: Algorithms and flowcharts	8
Worksheet 2: Algorithms and Flowcharts.....	9
Homework 2: Selection	13
Worksheet 3: Iteration	15
Homework 3: Iteration.....	17
Worksheet 4: Data types and arrays.....	21
Homework 4: Data types and arrays.....	25
Worksheet 5: Testing	27
Homework 5: Sound.....	33
Worksheet 6 Compression	36
Unit 8 System Design and Databases Answers.....	39
Worksheet 1 Computer Systems.....	39
Homework 1 Computer Systems.....	43
Worksheet 2 Top-down Design	46
Homework 2 Top-down Design.....	46
Worksheet 3 Validation and Verification.....	46
Homework 3: Validation and Verification	50
Worksheet 4 Designing a database table	54
Homework 4 Designing a database table.....	54
Worksheet 5: Querying a database.....	55
Homework 5: Querying a database	57
Relevant Quizlet Flashcards	60

Unit 7 Algorithms and Flowcharts **Answers**

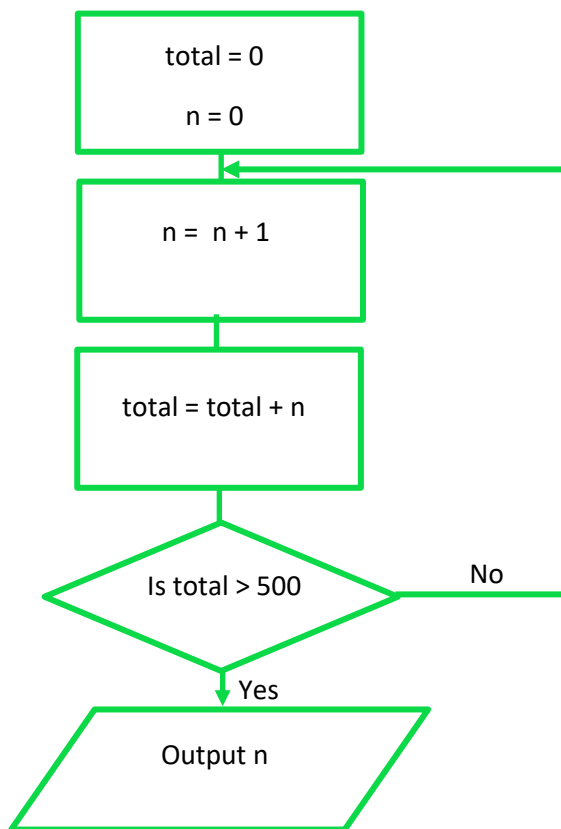
Worksheet 1: Algorithms and flowcharts

Task 1

1. Draw a flowchart for an algorithm which calculates how much money a student will need per week to buy a meal and two drinks each weekday. The user should be prompted to enter how much a meal costs, how much a drink costs, and then calculate and display the total required.



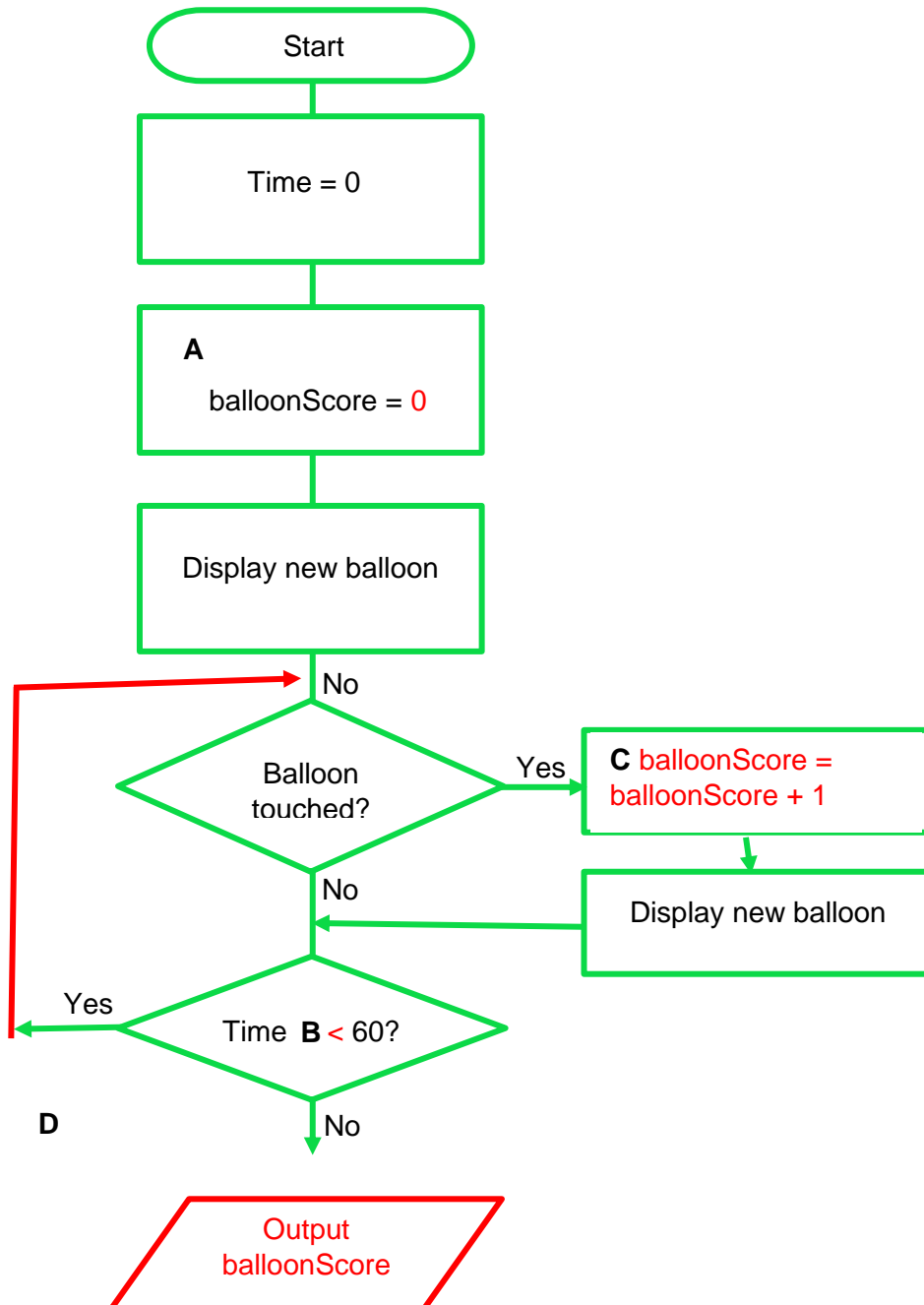
2. Draw a flowchart for an algorithm which calculates how many numbers 1, 2, 3...n have to be added to reach a total greater than 500. Output the answer n.



Task 2

Jasmine plays a game on her computer screen. A moving balloon appears on the screen, and she has to pop the balloon by clicking on it with the mouse. When the balloon is popped another one appears. The aim of the game is to pop as many balloons as possible in one minute.

The flowchart for the game is shown below.

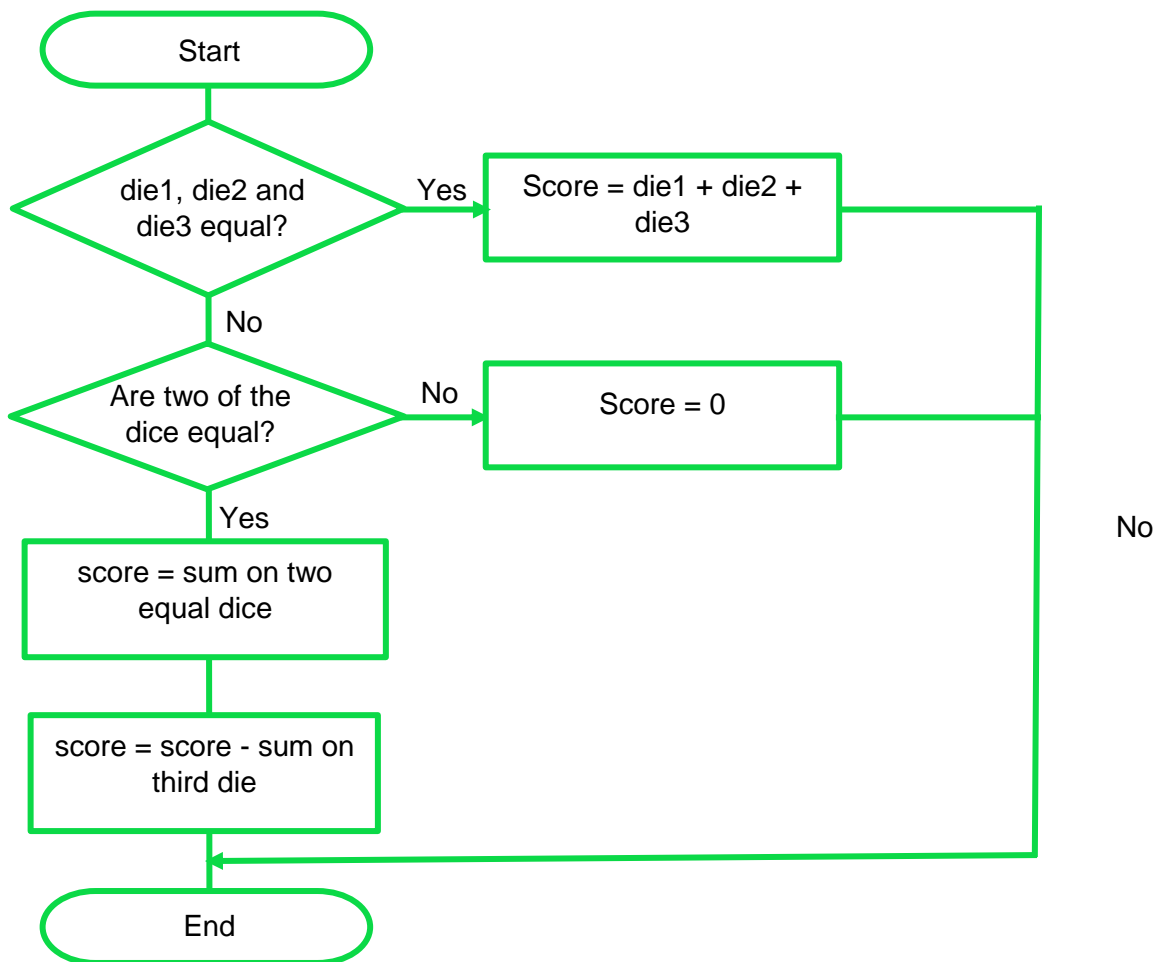


- (a) Complete the statement at A. **balloonScore = 0**
- (b) Complete the statement in B. **Time < 60**
- (c) Complete the statement at C.
- (d) Complete connector D.
- (e) Add a flowchart box at the bottom of the flowchart to display the player's score

Task 3

Michael is writing a program for a dice game played with three dice.

- (a) The player rolls the dice, and is given points according to the following algorithm.



State the value of the scores if the dice rolled are:

2 4 6 **0**

5 5 1 **9**

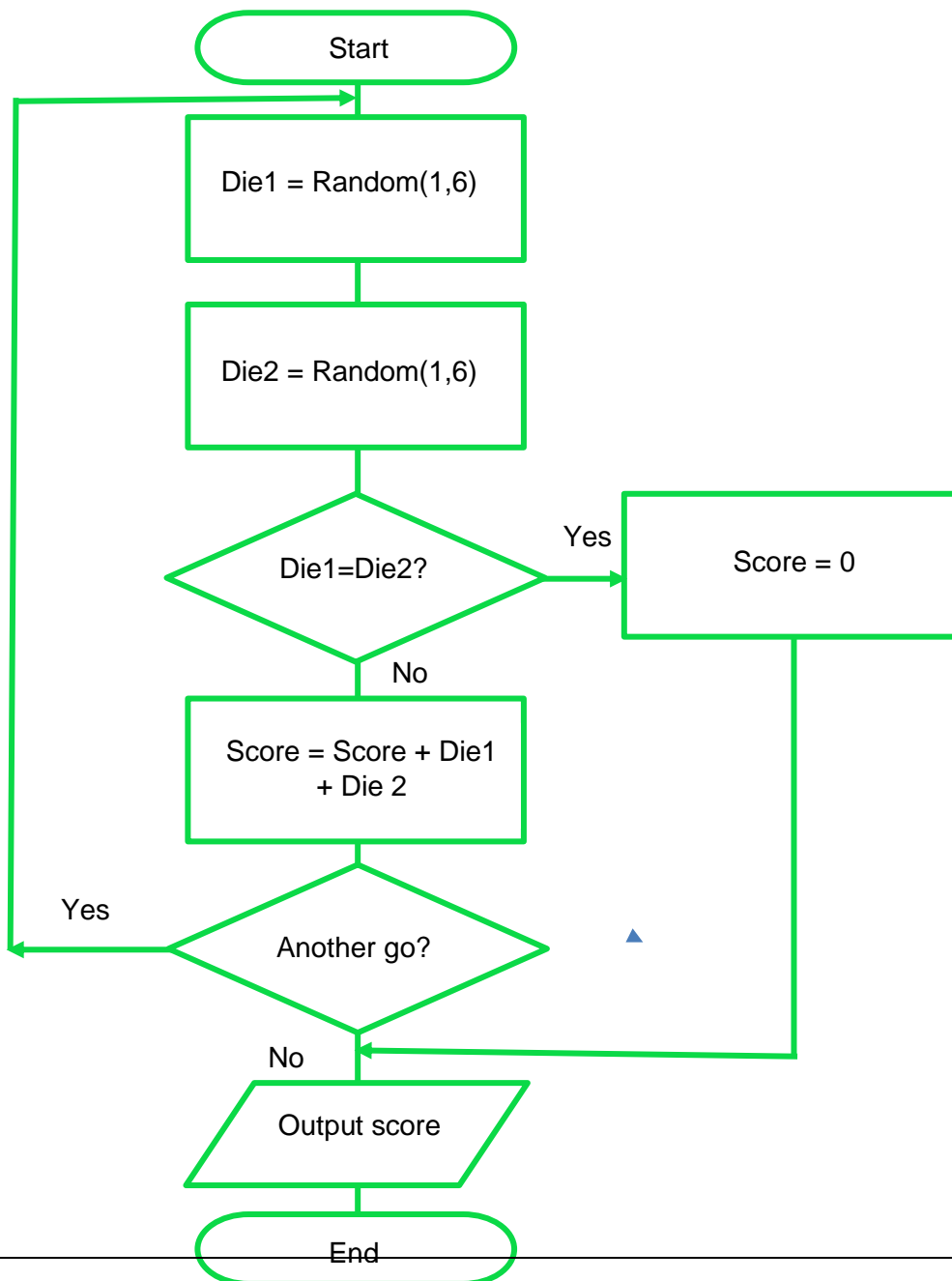
- (b) Some rolls of the dice produce a negative score. State a set of three numbers that can be used to test whether the algorithm produces a negative score when it should, and state the expected output of your test data.

115 or any answer where the sum of two dice is less than the third. Expected output -3

Task 4

James is writing a program to simulate a dice game. The function Random (1,6) generates a number between 1 and 6. He has drawn a flowchart to represent the algorithm to calculate a player's score when it is their turn. Paul and Coleen play the game.

- (a) Describe the rules of the game. **Each player can throw the dice as many times as they like, but as soon as they throw a double, their score reverts to zero and their turn ends. (Players take it in turns to throw the dice until one reaches a given target, but this is not shown on the flowchart, which represents a single turn.)**



- (b) Paul rolls the dice 3 times, getting six and two on the first throw, one and four on the second throw and two and three on the third throw. Coleen also rolls the the dice three times, getting five and six on the first throw, four and six on the second throw, two and two on the third throw.

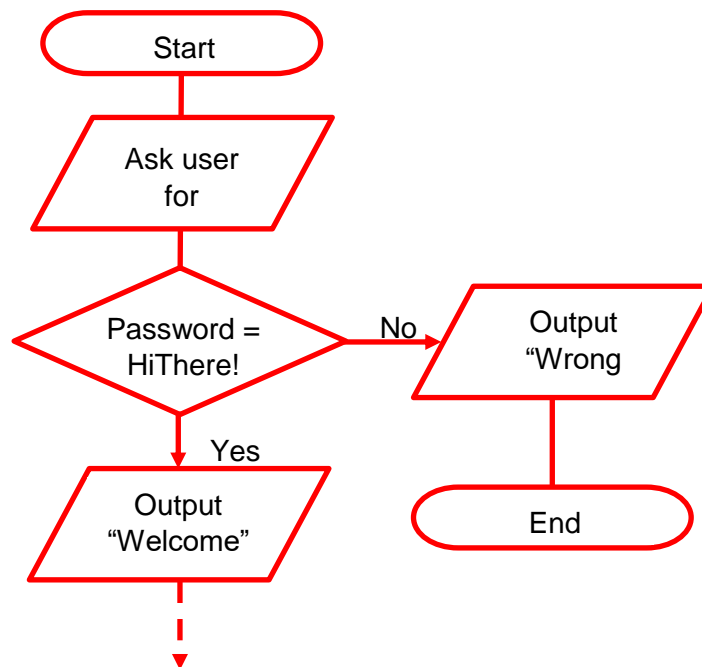
What are the scores of each player? **Paul's score is $8+5+5 = 18$**

Coleen's score is zero because she threw a double on her third throw. (She would have done better to stop after 2 throws).

Homework 1: Algorithms and flowcharts

Draw a flowchart for a program which asks the user to enter a password. If the user enters "HiThere!" then print "Welcome", and continue (indicate continuation with a dotted line).

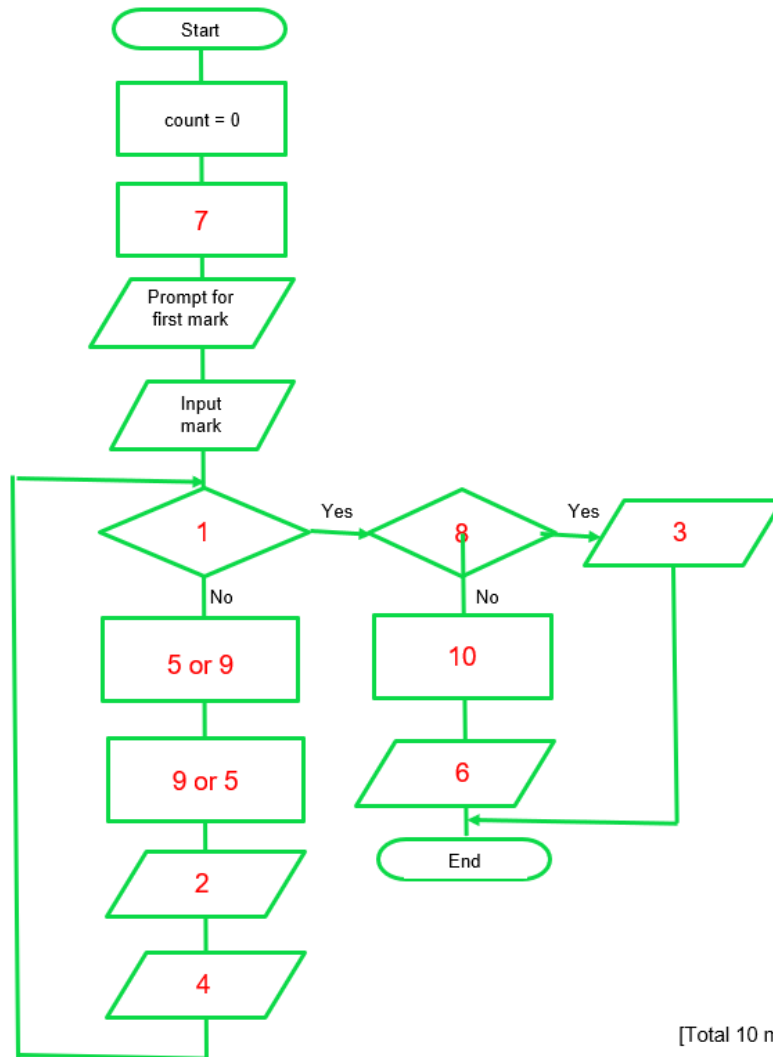
If they enter a different password, print "Wrong password" and end the program. [4]



2. An algorithm is to be written which calculates the average of a set of student marks entered at a keyboard. The flowchart on the next page describes the steps in the algorithm.

Item number	Item description
1	Is mark = -1?
2	Prompt for next mark
3	Output "No marks entered"
4	Input mark
5	Add 1 to count

6	Output average
7	total = 0
8	Is count = 0?
9	Add mark to total
10	average = total/count



[6]

[Total 10 marks]

Worksheet 2: Algorithms and Flowcharts

Task 1

1. Evaluate the following expressions, given that $a = 3$, $b = 4$ and $c = 12$

(i) $(a > b) \text{ OR } (b < c)$ **TRUE**

(ii) $(b > c) \text{ OR } (b > a)$ **TRUE**

(iii) $\text{NOT } (a > b)$ **TRUE**

(iv) (NOT (a>b)) AND (c > b) **TRUE**

(v) (a * b > c) AND (a = c / b) **FALSE**

2. Write pseudocode for a program which ask a multiple choice quiz question, e.g. "What is the world's largest ocean?". It then gives 3 possibilities numbered 1, 2 and 3 and asks the user to enter 1, 2 or 3. An appropriate response is then output.

```
OUTPUT ("What is the world's largest ocean?")
```

```
OUTPUT ("1. Atlantic")
```

```
OUTPUT ("2. Pacific")
```

```
OUTPUT ("3. Indian")
```

```
OUTPUT ("Answer 1, 2 or 3: ")
```

```
answer = USERINPUT
```

```
IF answer = 2 THEN
```

```
    OUTPUT ("Correct!")
```

```
ELSE
```

```
    OUTPUT ("No, it's the Pacific")
```

```
ENDIF
```

(see program in Python)

```
#multiple choice quiz question
```

```
print("What is the world's largest ocean? ")
```

```
print("1. Atlantic\n2. Pacific\n3. Indian")
```

```
answer = input("Answer 1, 2 or 3: ")
```

```
if answer == "2":
```

```
    print ("Correct!")
```

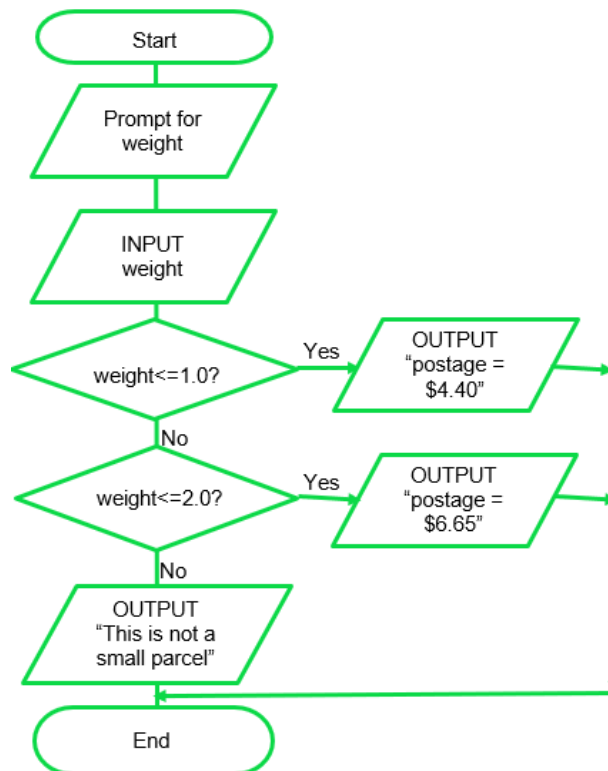
```
else:
```

```
    print ("No, it's the Pacific")
```

3. The cost of posting a small parcel of up to 1kg is is \$4.40. The cost for a parcel weighing more than 1kg, but less than or equal to 2kg, is \$6.55.

Draw a flowchart to allow the user to input the weight of the parcel and calculate and output the cost of postage. If the user enters a weight greater than 2kg, display a message "This is not a small parcel" and end the program.

Write the equivalent pseudocode for this algorithm.



Pseudocode:

```
OUTPUT ("Please enter weight of parcel")
weight = USERINPUT
IF weight <=1.0 THEN
    OUTPUT ("Postage = $4.40")
ELSE
    IF weight <=2.0
        OUTPUT ("Postage = $6.55")
    ELSE
        OUTPUT ("This is not a small parcel")
    ENDIF
ENDIF
ENDIF
```

4. Complete the table below to show 4 sets of test data you could use to thoroughly test your algorithm, and the expected result each time. Do not include invalid input data, since this section of pseudocode does not include a validation routine.

Test number	Parcel weight	Expected output
1	0.5	Postage = \$4.40
2	1.0	Postage = \$4.40
3	1.9	Postage = \$6.55
4	2.0	Postage = \$6.55
5	2.5	This is not a small parcel

Test data should include weights in each band and weights on the boundaries

Task 2

5. Nazim wants to create a game of scissors, paper, stone. In this game two people simultaneously make their hand into a shape representing scissors, paper or stone. Scissors cuts paper, so wins. Paper wraps stone, so wins. Stone breaks scissors, so wins.

Nazim starts by generating 2 random numbers between 1 and 3 to represent the three possibilities, and assigning them to scissors (1), paper (2) or stone (3).

- (a) Complete the program to print out what each player displayed, (scissors, paper or stone) and "Player1 wins" or "Player 2 wins" as appropriate.

```
Hand1 = random(1,3)
```

```
Hand2 = Random(1,3)
```

```
Case Hand1 of
```

```
1: Player1 = "Scissors"
```

```
2: Player1 = "Paper"
```

```
3: Player1 = "Stone"
```

```
End Case
```

```
Case Hand2 of
```

```
1: Player2 = "Scissors"
```

```
2: Player2 = "Paper"
```

```
3: Player2 = "Stone"
```

```
End Case
```

```
IF Player1 = "Scissors" THEN
```

CASE Player2 of:

"Scissors": Print ("Player 1", Player1, "Player2", Player2, "No one wins")

"Paper": Print ("Player 1", Player1, "Player2", Player2, "Player1 wins")

"Stone": Print ("Player 1", Player1, "Player2", Player2, "Player 2 wins")

ENDCASE

ENDIF

(other similar statements are required for Player1 = 2 or 3)

Could also use

CASE Hand2 of:

1: Print ("Player 1", Player1, "Player2", Player2, "No one wins")

2: Print ("Player 1", Player1, "Player2", Player2, "Player1 wins")

3: Print ("Player 1", Player1, "Player2", Player2, "Player 2 wins")

ENDCASE

Homework 2: Selection

1. Evaluate the following expressions, given that $x = 5$, $y = 6$, $z = 7$

(i) $(z \geq y) \text{ AND } (x \geq y)$ **FALSE**

(ii) $(y > x) \text{ OR } (y > z)$ **TRUE** [2]

2. An online bookstore gives free 2nd class mail delivery (code 2) for any order value greater than or equal to £15.00

For order values less than £15, 2nd class mail delivery costs £3.50.

For any value of order, a customer may choose to pay £5.00 for guaranteed next day delivery (code 1).

(a) Write pseudocode for an algorithm which allows the user to enter the total value of their order. They are then asked whether they want to pay for guaranteed next day delivery. Depending on their answer, and the value of the order, the program displays the postage charge and the overall total charge. [6]

OUTPUT ("Enter order value: ")

orderVal = USERINPUT

postageCharge = 5.00

OUTPUT ("Do you want to pay £5.00 for next day delivery? ")

OUTPUT ("Enter 1 for next day delivery, 2 for 2nd class post ")

```
postageCode = USERINPUT
IF orderVal >=15 and postageCode = 2 THEN
    postageCharge = 0
ELSE IF orderVal <15 and postageCode = 2 THEN
    postageCharge = 3.50
ENDIF
totalCharge = orderVal + postageCharge
OUTPUT (postageCharge, totalCharge)
```

(b) What will be the postage cost in each of the following cases?

(i) Order value £10.00 Postage code 2: **£3.50**

(ii) Order value £15.00 Postage code 2: **£0.0**

(iii) Order value £30.00 Postage code 1: **£5.00** [3]

2. Write pseudocode for one or more selection statements to decide whether a year is a Leap year. The rules are:

A year is generally a Leap Year if it is divisible by 4, except that if the year is divisible by 100, it is not a Leap year, unless it is also divisible by 400. Thus 1900 was not a Leap Year, but 2000 was a Leap year.
[4]

```
Year ← USERINPUT
LeapYear ← FALSE
IF (mod(Year, 4)=0) THEN
    LeapYear ← TRUE
ENDIF
IF (mod(Year,100) = 0 THEN
    LeapYear ← FALSE
ENDIF
IF (mod (Year,400)) = 0 THEN
    LeapYear ← TRUE
ENDIF
```

There are other variations – the order of the IF statements is important.

Worksheet 3: Iteration

The following pseudocode prints out a pattern. In this pseudocode, the statement

Write ("*") will print a star without moving to a new line. The statement
WriteLine will move to the start of a new line.

```
1.star = 1
2. While star <5
3.   For n = 1 to star - // n is new number input
4.     Write ("*")
5.   End For
6.   WriteLine
7.   Add 1 to star
8. End While
```



```
*
**
***
****
```

- (a) Draw in the box beside the code the pattern that this program produces. [3]
- (b) Give the line numbers of two lines that represent the programming construct "Sequence" **6 and 7**
[1]
- (c) Give the line numbers of two lines that represent the start of the programming construct "Iteration" **2 and 3** [1]
- (d) Write some pseudocode for a program that prints 5 lines in the pattern shown below.

```
X
XXX
XXXXX
XXXXXXX
XXXXXXXXX
```

[3]

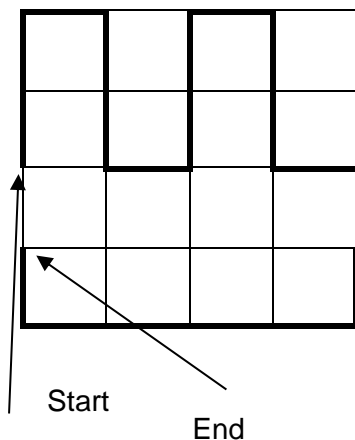
```
star = 1
While star < 10
    FOR n = 1 TO star
        Write ("X")
    NEXT
    WriteLine
    Add 2 to star
End While
```

2. A floor turtle uses these instructions.

Instruction	Meaning
FORWARD d	Move d cm forward
BACKWARD d	Move d cm backwards
LEFT t	Turn left t degrees
RIGHT t	Turn right t degrees
REPEAT n	Repeat the next set of instructions n times
ENDREPEAT	End of REPEAT loop
PENUP	Raise the pen
PENDOWN	Lower the pen

(Each square in the drawing is 10cm by 10cm)

(a) Complete the set of instructions to draw the shape shown in bold lines, starting at the point shown. [6]



```

PENDOWN
REPEAT 2
    FORWARD 20
    RIGHT 90
    FORWARD 10
    RIGHT 90
    FORWARD 20
    LEFT 90
    FORWARD 10
ENDREPEAT
PENUP

```


RIGHT 90
FORWARD 10
RIGHT 90
FORWARD 40
RIGHT 90
FORWARD 10

(There are other acceptable alternatives)

3. Write a pseudocode algorithm which asks a user to enter a number between 5 and 20. If they enter a number outside this range, the program asks them repeatedly to re-enter the number until they enter a valid number.

If a valid number n is entered, the program asks the user to enter n temperatures (all integer values), then calculates and outputs the average temperature. [6]

```
prompt for number between 5 and 20
n ← USERINPUT
WHILE n < 5 OR n > 20
    prompt for number between 5 and 20
    n ← USERINPUT
ENDWHILE

# enter temperatures and calculate average
total ← 0
FOR count = 1 to n
    prompt for temperature
    temperature ← USERINPUT
    total ← total + temperature
NEXT count
averageTemp = total/n
OUTPUT average
```

Homework 3: Iteration

Task 1

1. Examine the pseudocode program given below. The function $\text{mod}(a,b)$ gives the remainder when integer a is divided by integer b .

(a) Which lines show an example of the 'Sequence' programming construct?

Lines 1-2, 4-8, 11-12, 15-17

- (b) Which lines show an example of the 'Selection' programming construct? **10-13**
- (c) There are two examples of iteration statements in the program. On which lines do each of the 'Iteration' programming construct begin and end? **Lines 3 and 18, Lines 9 and 14**
- (d) If the user enters 1 and 10 for the first and last numbers in the range, what will be printed out at line 15? Which numbers is the program counting?

"7 numbers" (i.e. the number of integers between 1 and 10 not divisible by either 5 or 7)

```
1.  OUTPUT ("This program prints selected numbers in in given range.")
2.  AnotherGo ← "Yes"
3.  While AnotherGo = "Yes"
4.      OUTPUT ("Please enter the first number in your chosen range.")
5.      LowNumber = USERINPUT
6.      OUTPUT ("Please enter the last number in your chosen range.")
7.      Input (HighNumber)
8.      X ← 0
9.      For Count = LowNumber to HighNumber
10.         If (mod(count,5)<>0) AND (mod(count,7)<>0) then
11.             OUTPUT (Count)
12.             X ← X + 1
13.         End If
14.     Next count
15.  OUTPUT (X, "numbers")
16.  OUTPUT ("Another go?")
17.  AnotherGo = USERINPUT
18. Endwhile
```

Task 2

2. Write a pseudocode algorithm which inputs numeric scores and outputs how many of them are over 100. The end of the data is signalled by a user input of -1.

```
over100 ← 0
score ← 0
WHILE score <> -1
    OUTPUT ("Please enter the next score, -1 to end: ")
    score ← USERINPUT
    IF score > 100 THEN
        over100 ← over100 + 1
    ENDIF
ENDWHILE
```

Normally it is necessary to have the input statement at the end of the loop, so that it is tested immediately, as in the following example.

3. Write a pseudocode algorithm which inputs numeric scores and outputs the average score. The end of the data is signalled by a user input of -1.

```
OUTPUT ("Please enter the next score, -1 to end: ")
score = USERINPUT
totalScore ← 0
numScore ← 0
WHILE score <> -1
    numScore ← numScore + 1
    totalScore = totalScore + score
    OUTPUT ("Please enter the next score, -1 to end: ")
    score = USERINPUT
ENDWHILE
average ← totalScore / numScore
OUTPUT averageScore
```

To allow for the user entering -1 before entering any scores, the last two statements could be altered to:

```
IF numScore = 0 THEN
    OUTPUT ("Program ended")
ELSE
    average ← totalScore / numScore
    OUTPUT averageScore
```

ENDIF

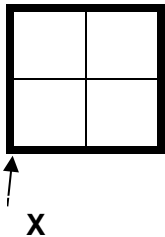
Task 3

4. A floor turtle uses these instructions.

Instruction	Meaning
FORWARD d	Move d cm forward
BACKWARD d	Move d cm backwards
LEFT t	Turn left t degrees
RIGHT t	Turn right t degrees
REPEAT n	Repeat the next set of instructions n times
ENDREPEAT	End of REPEAT loop
PENUP	Raise the pen
PENDOWN	Lower the pen

(Each square in the drawing is 10cm by 10cm)

- (a) Complete the set of instructions to draw the shape shown below in bold lines, starting at the point marked X.



PENDOWN

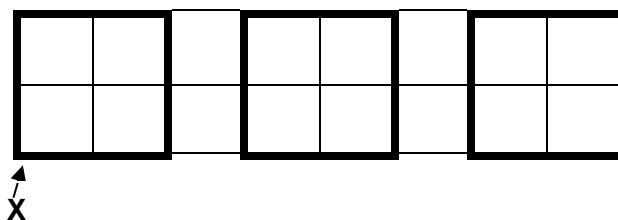
REPEAT 4

FORWARD 10

RIGHT 90

ENDREPEAT

- (b) Complete the set of instructions to draw the shape shown below in bold lines, starting at the point marked X.

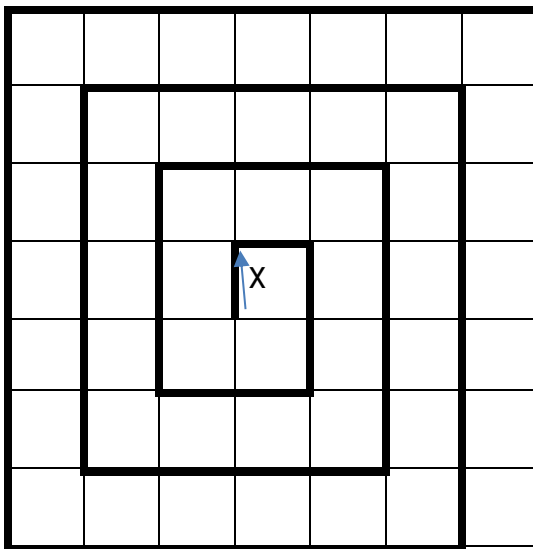


```

PENDOWN
REPEAT 3
  REPEAT 4
    FORWARD 10
    RIGHT 90
  ENDREPEAT
  PENUP
  RIGHT 90
  FORWARD 30
  LEFT 90
  PENDOWN
ENDREPEAT

```

- (c) Complete the set of instructions to draw the shape shown above in bold lines, starting at the point marked X



```

PENDOWN
distance 10
REPEAT 7
  REPEAT 2
    FORWARD distance
    RIGHT 90
  ENDREPEAT
  distance distance + 10
ENDREPEAT

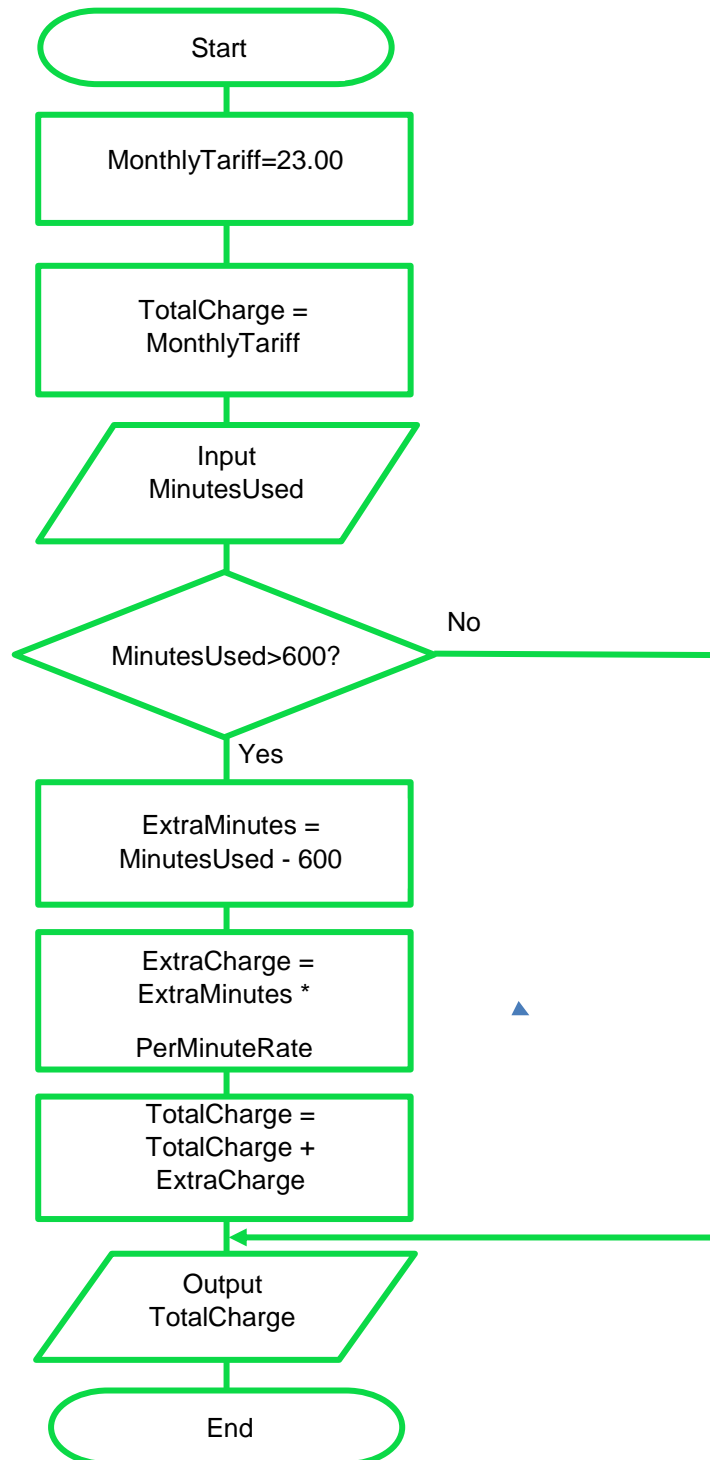
```

. A good way of working out the algorithm is to write it out without a loop, and then see how some instructions can be put inside a loop.

Worksheet 4: Data types and arrays

Task 1

The flowchart depicts an algorithm for computing the monthly bill for calls on a mobile phone. The tariff is as follows:



- (a) Explain how the total monthly charge is calculated.

There is a basic charge per month (MonthlyTariff). An extra charge is made for each minute used over 600 minutes in the month.

- (b) State whether each of the following variables should be declared as a variable or a constant, and what data type they should be given. Give reasons for your choice of data type in each case.

Identifier	Variable or constant?	Data type? Typical value?
MonthlyTariff	Constant	Real Typical value 20.00
TotalCharge	Variable	Real Typical value 25.50
MinutesUsed	Variable	Real Typical value 328.5
ExtraMinutes	Variable	Real Typical value 28.5
PerMinuteRate	Constant	Real Typical value 19.0
ExtraCharge	Variable	Real Typical value 3.55

- (b) Assume the basic monthly tariff is \$23.00 per month, and that extra minutes are charged at 19c per minute. A customer makes calls totalling 700 minutes on their phone. What is their total charge that month?

$$\text{\$23.00} + 100 * 0.19 = \text{\$42.00}$$

- (c) A second customer makes calls totalling 611.375 minutes. What is their total charge this month?

$$\text{\$23.00} + 11.375 * 0.19 = \text{\$23.00} + \text{\$2.16125} = \text{\$25.16125}$$

- (d) What function can you use to calculate and display the total charge to the nearest penny?

The ROUND function could be used. Typically, it would be called with a statement such as RoundedCharge = ROUND (TotalCharge, 2). The second parameter signifies that the answer should be rounded to 2 decimal places.

2. What value is returned in x by each of the following function calls?

- (a) `x = int (4.567)` 4
- (b) `x = round (345.862, 0)` 346
- (c) `x = float ("37.53")` 37.53
- (d) `x = sqrt (36)` 6

3. Write pseudocode for a program which simulate throwing two dice and then outputs the number on each die and the total of the two dice. The predefined function `random (x,y)` generates a random number between x and y.

```
die1 ← random(1,6)
die2 ← random(1,6)
total = die1 + die2
OUTPUT ("You threw a ", die1 "and a ", die2)
OUTPUT ("Total of the dice ", total)
```

Task 2

Julie has collected data on the average monthly temperatures for each month of the year. She is writing a program to print these out in a list, e.g.

```
January      15.0
February     13.8
March        16.5
etc.
```

You need to define an array `Months` ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]

She wants to allow the user to input the monthly rainfall into an array `rainfall[1..12]`.

Write pseudocode for inputting the data in Julie's program, prompting the user with statements such as "Enter rainfall for January", "Enter rainfall for February", etc., and then prints the data in a list.

```
FOR i = 1 to 12
    OUTPUT ("Please enter temperature for ", month[i])
    temperature [i] ← userinput
NEXT i
```

```
FOR i = 1 to 12
    OUTPUT (month[i], temperature[i])
NEXT i
```


Extension (Solution written as Python program)

Write pseudocode to define the array, input the monthly rainfall and print

- a list of each month's rainfall
- the total annual rainfall to one decimal place
- the monthly average over a year

the number of months that have rainfall above the average value.

```
Months = ["January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"]
rainfall = [0,0,0,0,0,0,0,0,0,0,0,0]
annualRainfall = 0
for i in range (0,12):
    print("Please input rainfall for month ",Months[i])
    rainfall [i] = int(input())
    annualRainfall = annualRainfall + rainfall[i]
print ("Thanks")
averageRainfall = annualRainfall/12
roundedAverage = round(annualRainfall/12,1)
monthsAboveAverage = 0
for i in range(0,12):
    if rainfall[i] > averageRainfall:
        monthsAboveAverage +=1 #add 1 to monthlyAverage
    print(Months[i], rainfall[i])
print("Annual rainfall = ", annualRainfall)
print ("Average monthly rainfall", roundedAverage)
print("\nNumber of months above average rainfall: ", monthsAboveAverage)
```

Homework 4: Data types and arrays

The following pseudocode tests how quickly a user can type a given sentence without making any mistakes. The third character of the sentence, for example, is referred to as sentence[2].

The statement finishTime = time.clock() returns wall-clock seconds elapsed since the first call to this function and places the answer in finishTime.

1. import time
2. sentence ← "The quick brown fox jumped over the lazy dog"
3. n ← len(sentence)
4. OUTPUT ("Number of characters",n)
5. firstChar ← sentence[0]
6. OUTPUT ("First character is ", firstChar)
7. errorFlag ← 0
8. OUTPUT ("Start typing! Press Enter when finished ")
9. startTime ← time.clock()

```

10. mySentence ← USERINPUT
11. finishTime ← time.clock()
12. OUTPUT ("Sentence to type: ", sentence)
13. OUTPUT ("You typed: ", mySentence)

14. IF mySentence <> sentence THEN
15.     errorFlag ← 1
16. ENDIF
17. totalTime ← finishTime
18. startTime ← round(startTime,3)
19. finishTime ← round(finishTime,3)
20. OUTPUT ("Start time, finish time ", startTime, finishTime)
21. totalTime ← round(totalTime,3)
22. IF errorFlag THEN
23.     OUTPUT ("You made one or more errors")
24. ELSE
25.     OUTPUT ("Total time taken ",totalTime, "seconds")
26. ENDIF

```

(a) What does the statement import time do?

[1]

It imports the time module which contains pre-written functions which can be imported into the user's program.

(b) What type of variable is each of the following?

[4]

- (i) errorFlag **Boolean**
- (ii) n **integer**
- (iii) totalTime **real or float**
- (iv) mySentence **string**

(c) What will be output at line 6? **T**

[1]

(d) What does line 21 do?

[1]

It rounds the total time to 3 decimal places

(e) Alter the program so that instead of storing the sentence "The quick brown fox jumped over the lazy dog", the user can enter the sentence on which they will be timed.

[2]

OUTPUT ("Type a sentence to be timed on")

INPUT (sentence)

2. Sara is writing a program to input her monthly phone bills and output the month name and amount for the month with maximum amount.

She has defined an array to hold the month names.

Complete the pseudocode program.

[6]

```
# Program to output maximum month's phone bill
```

```
MonthName ["January", "February", "March", "April", "May", "June", "July", "August", "September",  
"October", "November", "December"]
```

```
# Define an array to hold the phone bills for each month
```

```
PhoneBill ← [0,0,0,0,0,0,0,0,0,0,0,0]
```

```
MaxPhoneBill ← 0
```

```
FOR i = 1 TO 12
```

```
  OUTPUT("Please input phone bill for month ", MonthName[i])
```

```
  PhoneBill[i] ← USERINPUT
```

```
  IF PhoneBill[i] > MaxPhoneBill THEN
```

```
    MaxPhoneBill = PhoneBill[i]
```

```
    MaxMonth = MonthName[i]
```

```
  ENDIF
```

```
NEXT i
```

```
OUTPUT ("Maximum Phone Bill: ",MaxMonth, MaxPhoneBill)
```

Worksheet 5: Testing

Task 1

1. The pseudocode program shown below is intended to model the calculation of the toll charge for crossing a bridge on a motorway. Between 6am (0600hrs) and 10pm (2200 hrs) inclusive, the charges are as follows:

Motorcycle	\$1.00	(Vehicle type 1)
Car	\$2.00	(Vehicle type 2)
Goods vehicle	\$2.50	(Vehicle type 3)
HGV and coaches	\$5.00	(Vehicle type 4)

After 10pm, up to but not including 6am, there is no charge.

Amanda writes the following pseudocode:

```
IF time <=0600 OR time >=2200 THEN
```

```
  Charge = 0
```

```
ELSE
```

```
  CASE Vehicle OF
```

```
    1: Charge ← 1.00
```

```

        2: Charge ← 2.00
        3: Charge ← 2.50
        4: Charge ← 5.00
    ELSE
        Charge ← 0.00
    ENDCASE
ENDIF

```

Complete the following test plan. You can assume that the variable time has already been validated and is between 0000 and 2400 (24-hour clock).

Test number	Data (Vehicle type)	Data (Time)	Reason for test	Expected result for Charge	Actual result
1	1	1245	Valid time	1.00	1.00
2	2	0600	Boundary time	2.00	0.00
3	3	2200	Boundary time, vehicle type 3	2.50	0.00
4	4	2201	Boundary time, vehicle type 4	0.00	0.00
5	5	0001	Time <0600, vehicle type 5	0.00	0.00

(any reasonable variations on these tests are acceptable)

Are your tests sufficient to thoroughly test the program? What assumptions have been made?

You could argue that these tests are not sufficient to test charges for all vehicles at different times. Add in tests for 0559 and 2201. You could test 0001, 2401, -1 and xxx (invalid) but if the clock is automatically read maybe the last three are not necessary! Do all vehicle types need to be tested for all times?

Is the logic of the program correct? If not, what should be changed? **Not correct...**

The condition needs to be changed to IF time <0600 OR time >2200 THEN

Task 2

The following program code is part of a game in which Mighty Max is engaged with the enemy, dealing deathly blows in all directions to overcome the evil gremlins but taking considerable punishment himself until he is fatally wounded.

```

wounds ← 0
gremlins ← 4
strength ← 30

```

```
WHILE strength > 0
```

```
  IF gremlins >= 1 THEN
```

```
    wounds ← wounds + gremlins
```

```
  ENDIF
```

```
  IF strength > 2 THEN
```

```
    gremlins ← gremlins - 1
```

```
    output "Mighty Max has dealt a deathly blow to a gremlin, but his strength is fading"
```

```
  ENDIF
```

```
  strength ← strength - wounds
```

```
  output "Wounds = ", wounds, "Gremlins = ", gremlins, " Strength = ", strength
```

```
ENDWHILE
```

```
output "Alas, our hero has been overcome... Game Over"
```

Complete the next 4 rows in the following trace table.

Wounds	Gremlins	Strength	Strength<>0
0	4	30	True
4	3	26	True
7	2	19	True
9	1	10	True
10	0	0	False

Task 3

When the creator of the game of chess showed his invention to the ruler of the country, the ruler was so pleased that he gave the inventor the right to name his prize for the invention. The man, who was very clever, asked the king this: that for the first square of the chess board, he would receive one grain of wheat (in some tellings, rice), two for the second one, four on the third one, and so forth, doubling the amount each time. (You can Google the full story, "grains of wheat on chessboard")

A programmer decides to find out how many grains of wheat the inventor will end up with. He tests the program with just 6 squares of the chessboard.

The program is shown below. Complete the trace table to show the total number of grains of wheat received for the first 6 squares of the chessboard.

OUTPUT ("How many grains of wheat will be on each square?")

powerOf2 = 1

total = 1

FOR n = 2 To 6

 powerOf2 = powerOf2 * 2

 total = total + powerOf2

 OUTPUT ("You have a total of ",total, "grains on squares 1 to",n)

NEXT n

PowerOf2	Total	n	OUTPUT
1	1	2	
2	3	3	
4	7	4	
8	15	5	
16	31	6	
32	63		

Task 4

- (a) The Post Office charges different rates for small, medium and large parcels.

Write a pseudocode algorithm to determine whether a parcel is small, medium or large. Set a variable called size equal to 's', 'm' or 'l' according to its size, and display a message to tell the user whether the parcel is small, medium or large.

A small parcel must not exceed: Length 45cm, Width 35cm, Depth 16cm, Weight 2kg.

A medium parcel must not exceed: Length 61cm, Width 46cm, Depth 46cm, Weight 20kg.

A parcel larger than this cannot be posted.

```
IF (length<=45) AND (width <=35) AND (depth <=16) AND (weight <=2) THEN
    size = 's'
ELSE
    IF (length<=61) AND (width <=46) AND (depth <=46) AND (weight <=20) THEN
        size = 'm'
    ELSE
        size = 'l'
    ENDIF
ENDIF
ENDIF
```

- (b) Complete the table below to show 6 sets of test data you could use to thoroughly test your algorithm, and the expected result each time. Do not include invalid input data, since this section of pseudocode does not include a validation routine.

Test number	Length	Width	Depth	Weight	Reason for test	Expected output (size)
1	20	20	5	1.2	Test "normal" data for small parcel	s
2	45	35	16	2	Test "boundary" data for small parcel	s
3	45	35	16	3	Test mixed dimensions for small and medium parcel	m
4	50	20	20	1	Test mixed dimensions for medium parcel	m
5	61	46	46	20	Test "boundary" data for medium parcel	m

6	62	10	10	2	Test one dimension too large for medium parcel	1
---	----	----	----	---	--	---

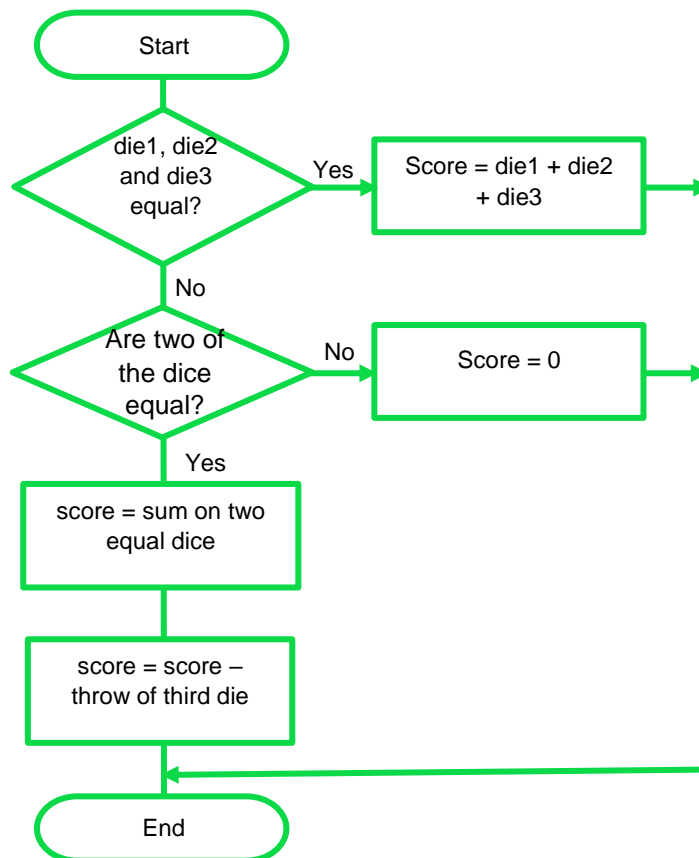
Does your test plan comprehensively test your algorithm?

No, you could still have had a wrong dimension in the pseudocode algorithm which would not be picked up e.g. if the width of a medium parcel had been wrongly typed in as 64 instead of 46.

Task 5

In Worksheet 1 Task 3 you were given the following algorithm for a dice game.

Write a test plan with test data and expected results.



In order to test the program you need to input the number on each die, rather than having it randomly generated.

Also, it is useful to insert a loop so that the dice are rolled for example 5 times without having to start a new run of the program each time.

The test plan should include rolls where no dice are equal, all dice are equal, and 3 combinations of equal dice are given, at least one of which gives a negative answer. Eight tests would comprehensively test all the options. (See program Worksheet 5 Task 5 test dice game1.py)

Test	Die1	Die2	Die3	Purpose of test	Expected result
1	5	5	5	All dice equal	15
2	1	2	3	No dice equal	0
3	1	1	6	Dice 1 and 2 equal, negative score	-4
4	5	5	2	Dice 1 and 2 equal, positive score	8
5	1	6	1	Dice 1 and 3 equal, negative score	-4
6	6	1	6	Dice 1 and 3 equal, positive score	11
7	5	2	2	Dice 2 and 3 equal, negative score	-1
8	5	6	6	Dice 2 and 3 equal, positive score	7

Homework 5: Testing

The following pseudocode calculates and outputs two values k and j. It then calculates and outputs two further values t and a.

```

x = [7, 9, 6, 2, 4]
t = x[1]
k = x[1]
j = x[1]
FOR n = 2 TO 5
    t = t + x[n]
    IF x[n] > k THEN
        k = x[n]
    ENDIF
    IF x[n] < j THEN
        j = x[n]
    ENDIF
NEXT n
OUTPUT ("k = ", k, "j = ", j)
t = t - k - j
a = t/3
a = int(a)
OUTPUT ("a = ", a)

```

(a) Complete the trace table below

t	k	j	n	x[n]	a	OUTPUT
---	---	---	---	------	---	--------

7	7	7	2	9		
16	9	7	3	6		
22	9	6	4	2		
24	9	2	5	4		
28	9	2				k = 9 j = 2
17					5	a = 5

[6]

(b) Describe what the program does.

[6]

The program finds the maximum and minimum numbers in an array x

It finds the total of all the numbers in the array

It then outputs the maximum and minimum number in the array (k and j)

Then it subtracts these values from the total, calculates the average of the remaining numbers in the array

Converts this to an integer value and outputs the result

(c) List two ways in which the program could be made easier to understand.

[2]

Add comments to say what sections of code are doing

Use meaningful variable names

2. A program produces statistics on the number of passes and distinctions in an exam, and the number of candidates who took the exam. A student achieving a Distinction is also counted among the number of students who passed the exam. The exam is graded as follows:

50-79 Pass

80-100 Distinction

The program has been written in pseudocode, but has errors in it.

The programmer tests it by inputting the exam results 60, 65, 70, 90, 95, 100. The end of data entry is signified by entering a dummy mark of -1.

The program outputs the following results:

Number of students: 7

Number of students with pass or distinction: 6

Number of students with distinction: 0

Average mark: 1.0

The pseudocode is given below:

```
mark = 0
studentPass = 0
distinction = 0
numStudents = 0
totalMark = 0
WHILE mark <> -1
    OUTPUT("Please enter mark, -1 to end: ")
    mark = USERINPUT
    IF mark >= 50 THEN
        studentPass = studentPass + 1
    ELSE
        IF mark >= 80
            distinction = distinction + 1
        ENDIF
    ENDIF
    numStudents = numStudents + 1
    totalMark = totalMark + 1
ENDWHILE
averageMark = totalMark/numStudents
OUTPUT ("Number of students: ", numStudents)
OUTPUT ("Number of students with pass or distinction: ", studentPass)
OUTPUT ("Number of students with distinction: ", distinction)
OUTPUT ("Average mark: ",averageMark)
```

Locate the two errors, and suggest how the coding should be corrected.

[6]

- Error 1 The first OUTPUT and USERINPUT statements should be placed before entering the loop. Within the loop, the OUTPUT and USERINPUT statements should be at the end of the loop.
- Error 2 There should not be an ELSE statement. The second IF statement
 "IF mark > 80
 distinction = distinction +1

ENDIF"

is a separate statement.

Error 3 `totalMark = totalMark + 1` should be
`totalMark = totalMark + mark`

Worksheet 6 Effectiveness of Algorithms

Task 1

1. (a) Multiply the decimal number 4567 by 100 **456700**

(b) What is the value of 10^4 ? **10000**

(c) Multiply 6540 by 10^4 **65400000**

(d) Describe the rule that you used to calculate the answer to (c)

To multiply by 10^n , add n noughts

(e) Multiply the binary number 1111 by 8, leaving the answer in binary. **1111000**

(f) Describe the method you used to do this.

$8 = 2^3$ To multiply by 2^3 , add 3 noughts

Task 2

2. Using the binary search algorithm, you should be able to guess a number between 1 and 2^n in $n+1$ guesses.

(a) What is the maximum number of guesses it will take to find a number between 1 and 64?

6 guesses

(b) What is the maximum number of guesses it will take to find a number between 1 and 1000? Explain your answer.

11 guesses since 1000 is between 2^9 and 2^{10}

- (c) On average, how many guesses would you need to find a number between 1 and 1000 if you performed a linear search – i.e., guess 1, 2, 3 etc until you find the number? What does this tell you about the efficiency of the binary search algorithm?

On average, 500. The binary search is extremely efficient, especially for a large number of items.

3. The following algorithm performs a binary search to find an item in a sorted array A of length n. If the item is found, its position in the array is displayed. If the item is not in the list, "Item not found" is displayed. Assume that the first element of the array is A[1].

```
itemFound ← False
searchFailed ← False
top ← n
bottom ← 1
REPEAT
    midpoint = int ((top + bottom)/2)
    IF A[midpoint] = itemSought THEN
        itemFound = True
    ELSE
        IF bottom > top THEN
            searchFailed = True
        ELSE
            IF A[midpoint] < itemSought THEN
                bottom = midpoint + 1
            ELSE
                top = midpoint - 1
            ENDIF
        ENDIF
    ENDIF
UNTIL itemFound OR searchFailed
IF itemFound THEN
    OUTPUT ("Item is at position", midpoint)
ELSE
    OUTPUT ("Item is not in the array")
```

The array contains the names

Anna, Bill, David, Faisal, Jasmine, Jumal, Ken, Michela, Pavel, Rosa, Stepan, Tom, Zac

- (a) The user searches for the name Tom. Complete the trace table below

itemFound	searchFailed	top	bottom	midpoint	A[midpoint]	OUTPUT
-----------	--------------	-----	--------	----------	-------------	--------

false	false	13	1	7	Ken	
			8	10	Rosa	Item is at position 10

(b) The user searches for the name Erik. Complete the trace table below

itemFound	searchFailed	top	bottom	midpoint	A[midpoint]	OUTPUT
false	false	13	1	7	Ken	
		6		3	David	
			4	5	Jasmine	
		4		4	Faisal	
		3	4	3		
	true					Item is not in array

Task 3

The simplest (and slowest) method of sorting an array of values into alphabetical or numerical sequence is the Bubble Sort. It works like this:

Suppose you have an array of 6 usernames: Carl, Tasmin, Eric, Zoe, Alan and Mark

Go through the list, comparing each name with the one next to it. If it is greater, swap them.

After going through the list once, it looks like this: Carl, Eric, Tamsin, Alan, Mark, Zoe.

The last element of the array is now in the correct place.

(i) Repeat the operation on the first 5 names. What is the order now?

Carl, Eric, Alan, Mark, Tamsin, Zoe

(ii) Show the order of the names after each further pass through the array until the names are sorted.

Carl, Alan, Eric, Mark, Tamsin, Zoe

Alan, Carl, Eric, Mark, Tamsin, Zoe

The pseudocode algorithm is:

```
userName ← ["Carl","Tamsin","Eric","Zoe","Alan","Mark"]
numItems ← 6
WHILE numItems>1
    FOR count = 1 to 5
        IF userName [count] > userName[count+1] THEN
            Swap the names in the array
        ENDIF
    NEXT count
    numItems ← numItems - 1
ENDWHILE
```

(iii) Write pseudocode statements to replace the statement *Swap the names in the array* to show how this operation will be performed.

```
temp ← username[count]
username[count] ← username[count+1]
username[count+1] ← temp
```

(iv) IF userName [count] < userName[count+1] THEN

Unit 8 System Design and Databases **Answers**

Worksheet 1 Computer Systems

Task 1

The computer system in a library has several subsystems.

- Stock control
- Loans
- Accounts
- Payroll

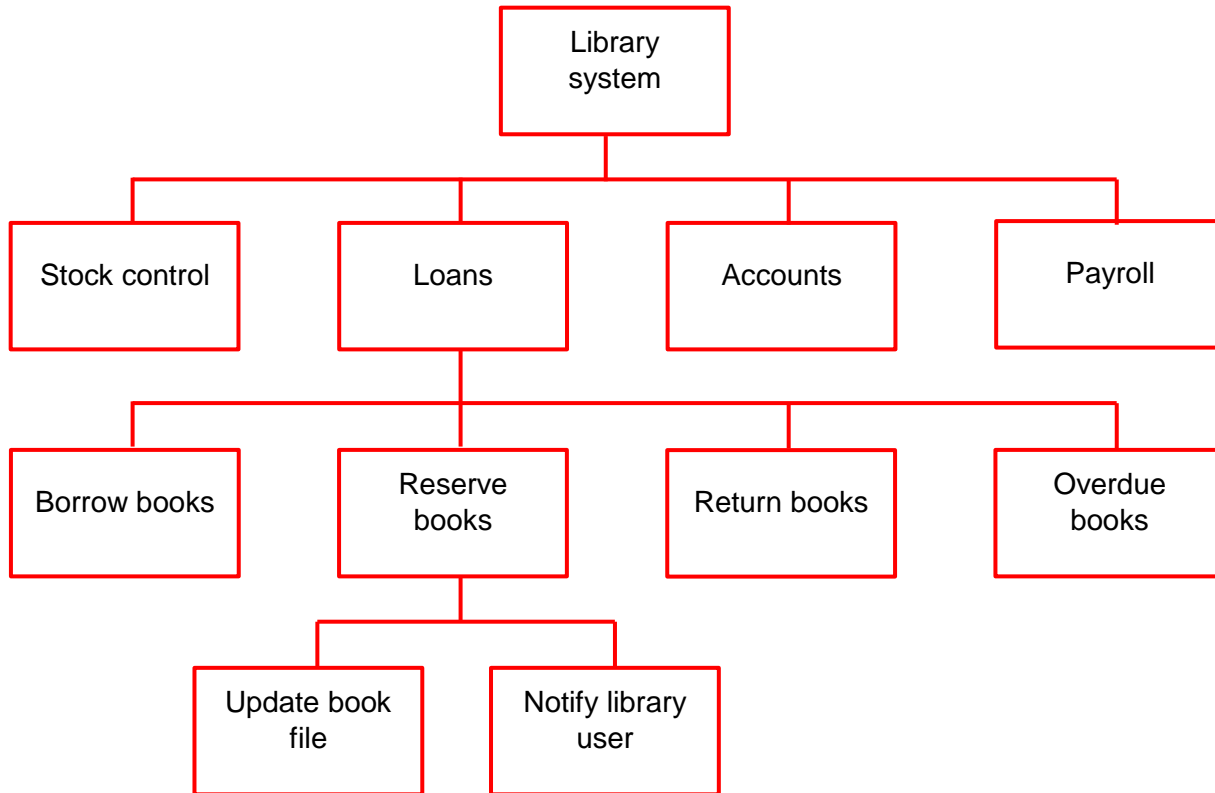
The loans sub-system has further sub-systems, performing the following functions:

- borrow books
- reserve books
- return books
- overdue books

The “reserve books” sub-system performs the following functions:

- update book file so that book cannot be borrowed by anyone else
- notify library user by email when book becomes available

Draw a structure chart showing the library system and sub-systems.



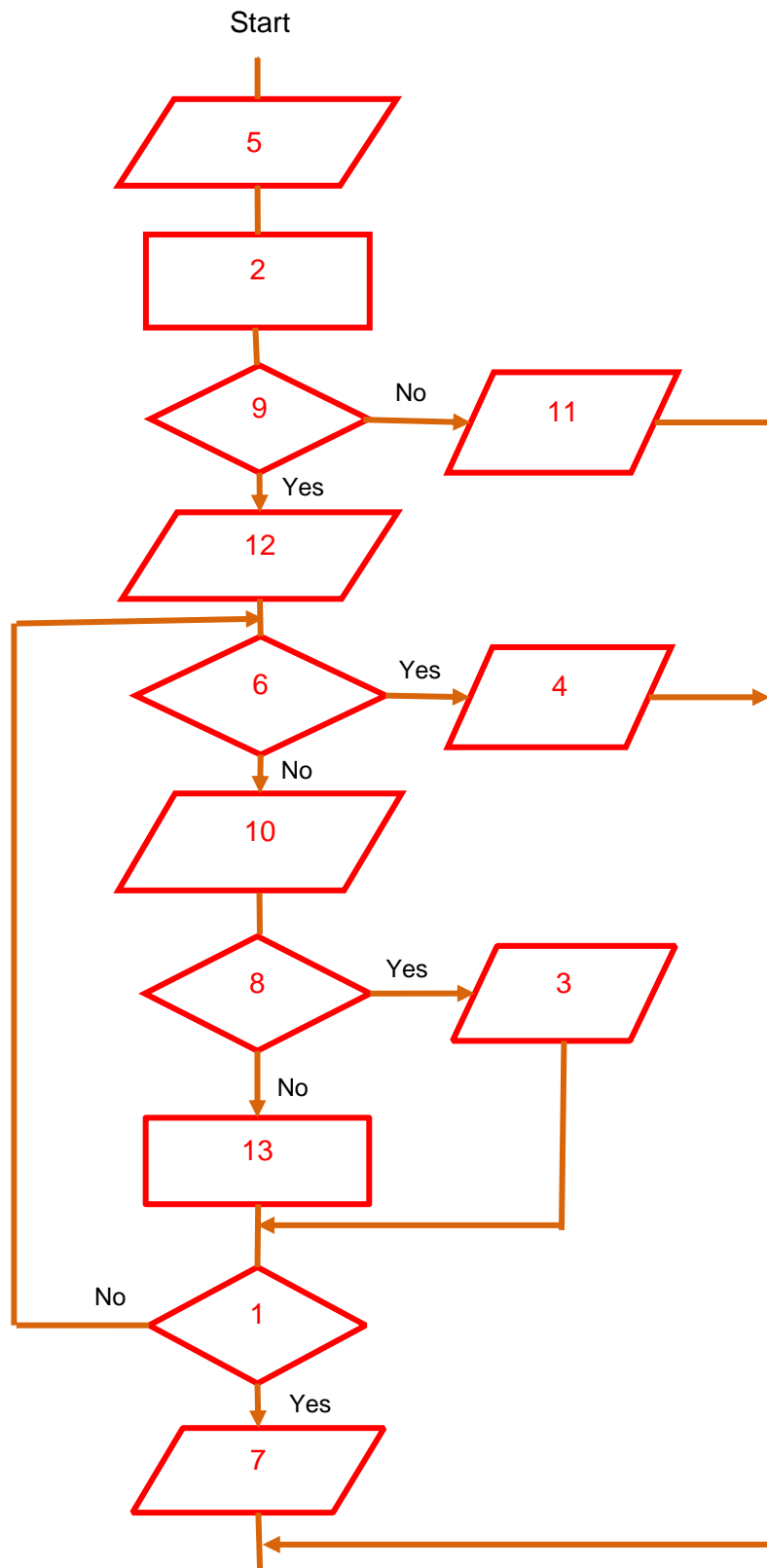
Task 2

The flowchart on the next page shows how the “Borrow books” sub-system works. Several statements are missing from the flowchart.

Complete the flowchart, **using item numbers only**, from the list below.

Item number	Description
1	Finished?
2	Search file for borrower information
3	Display message “This book is reserved”
4	Display message “You already have the maximum books on loan”

- | | |
|-----------|--|
| 5 | Scan barcode on borrower's card |
| 6 | Books on loan = maximum allowed? |
| 7 | Print slip listing titles and due dates for return |
| 8 | Is book reserved by another user? |
| 9 | Is the borrower ID in the file? |
| 10 | Scan barcode in book |
| 11 | Display error message "Invalid library card" |
| 12 | Display "Please remove card" |
| 13 | Update loans file with borrower and book details |



End

Homework 1 Computer Systems

What is meant by “top-down design”?

[3]

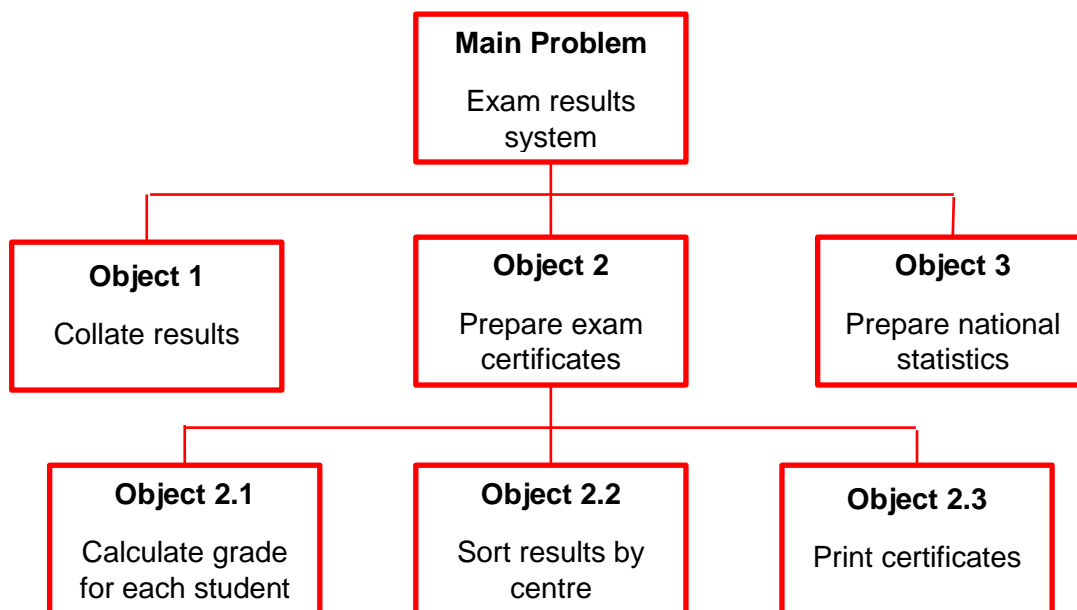
“Top-down design” means tackling a complex system by

- Breaking the system down into sub-systems
- Breaking these sub-systems into further sub-systems as necessary to achieve modules which each perform a single well-defined task

2. An exam board uses a computer system to record examination entries, record marks for each exam, and reporting on results for each subject.

Reporting on results involves collating results for each student in each centre (school or college) for which there are results, sending out certificates for each student showing exams passed and grades obtained, and preparing national statistics on how many students took each exam and the percentage of students gaining each grade in each subject.

Draw a structure diagram to illustrate how this computer system may be broken down into sub-systems, which in turn may be made up of further subsystems. [5]



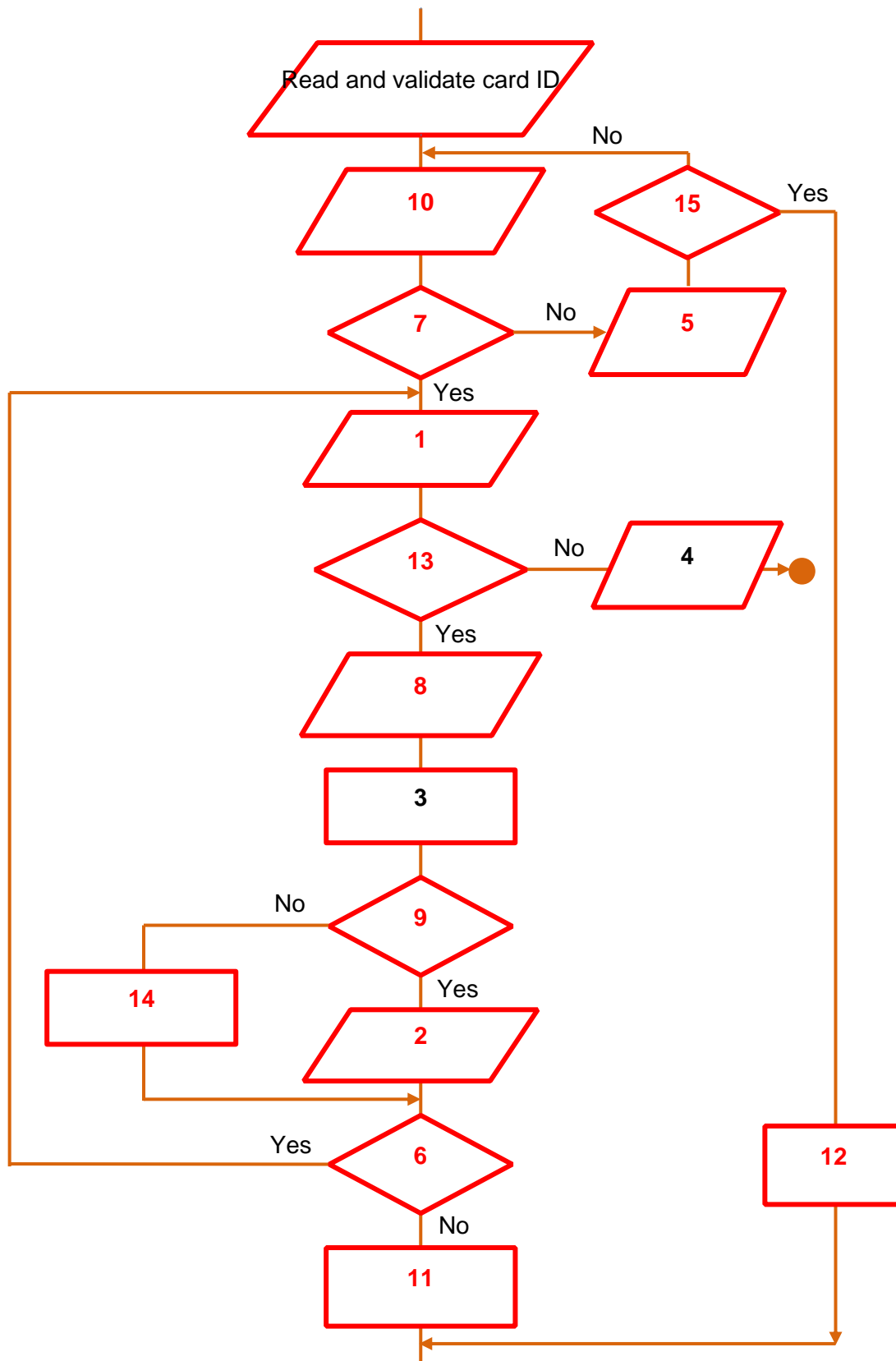
(Allow any reasonable looking breakdown of tasks, with 2 levels under the top level)

3. The flowchart on the next page shows part of a computer system which allows bank users to use an ATM (cash machine) to check their bank balances, withdraw cash or top up their mobile phones. Several statements are missing from the flowchart.

Complete the flowchart, using item numbers only from the list below.

[12]

Item number	Description
1	Display menu of options
2	Dispense cash
3	Check customer balance
4	Display other options
5	Display "PIN incorrect"
6	Customer wants to perform another transaction?
7	PIN correct?
8	Customer inputs amount to withdraw
9	Enough cash in customer account?
10	Input PIN
11	Eject card
12	Bank retains card
13	Customer wants to withdraw cash?
14	Display "Insufficient funds in account"
15	Has customer had 3 attempts to input PIN?



Worksheet 2 Top-down Design

Homework 2 Top-down Design

Worksheet 3 Validation and Verification

Task 1

1. Data on pupils registering at a secondary school needs to be entered. The data is as follows:

Registration number (6 numeric digits)

Surname

First name

Date of Birth

Gender

Phone number

Name and describe the validation checks that could be carried out on each of these fields.

Registration number: Length check (6 digits) and type check (integer)

Date of birth: Valid date Range check: should be within a certain range

Gender: Should be "M" or "F"

Phone number: Hard to check this as it may contain numbers, brackets, spaces, hyphens. It should be input as a string.

2. A product code is 7 digits long, including a check digit at the right hand end. The check digit is calculated as follows:
 - assign weights of 2, 3, 4, 5, 6, 7 to each of the digits of the number, starting from the rightmost digit
 - multiply each digit by its weight
 - add the resulting numbers together
 - divide by 10 using integer division
 - Subtract the remainder from 10
 - If the remainder is zero, the check digit is zero
 - Otherwise, subtract the remainder from 10 and the result is the check digit

(a) Calculate the check digit for the number 355416. What is the final product code?

3 5 5 4 1 6

7 6 5 4 3 2

$21 + 30 + 25 + 16 + 3 + 12 = 107/10 = 10 \text{ remainder } 7$

Check digit is $10 - 7 = 3$

Product code is 3554163

(b) Which of the following product codes is valid? Show your working.

5673110 5763110

$5 \times 7 + 6 \times 6 + 7 \times 5 + 3 \times 4 + 1 \times 3 + 1 \times 2 + 0 \times 1$

$= 35 + 36 + 35 + 12 + 3 + 2 + 0 = 123$

This is not divisible by 10 so the product code is invalid.

$5 \times 7 + 7 \times 6 + 6 \times 5 + 3 \times 4 + 1 \times 3 + 1 \times 2 + 0 \times 1$

$= 35 + 42 + 30 + 12 + 3 + 2 + 1 =$

3. (a) In December 2005, a Japanese securities trader made a \$1 billion typing error, when he mistakenly sold 600,000 shares of stock at 1 yen each instead of selling one share for 600,000 yen.

What validation check could have been applied to prevent this type of error?

There could be a range check on both the number of shares and the value of each share.

- (b) A Norwegian woman mistyped her account number on an internet banking system. Instead of typing her 11-digit account number, she accidentally typed an extra digit, for a total of 12 numbers. The system discarded the extra digit, and transferred \$100,000 to the (incorrect) account.

What validation checks could have been applied to prevent this type of error?

There could be a length check on the account number

Task 2

4. Use the trace table below and trace through the following program and record the output if the user enters the numbers (i) 6, 0, 4 (ii) 7 8, 9

OUTPUT "Please enter your choice (1-5)"

validChoice = "false"

count = 0

WHILE (validChoice = "false") AND (count<3)

 choice = USERINPUT

 count = count + 1

 IF choice < 1 OR choice > 5 THEN

 OUTPUT "Invalid choice"

 ELSE

 validChoice = "true"

 ENDIF

ENDWHILE

IF validChoice = "false" THEN

 OUTPUT "Program ending"

ELSE

 OUTPUT "continue"

ENDIF

validChoice	choice	count	OUTPUT
False	0		
False	6	1	Invalid choice
False	0	2	Invalid choice
False	4	3	continue

validChoice	choice	count	OUTPUT
False	0		
False	7	1	Invalid choice
False	8	2	Invalid choice
False	9	3	Program ending

5. (a) Write a pseudocode algorithm to ask a user to enter a new password. The algorithm then asks the user to re-enter the password, and if the two passwords match, the message "Please continue" is output. If they do not match, the program outputs a message "Incorrect password – please re-enter a new password", and loops until the user enters 2 identical passwords.

match = False

WHILE match = False

OUTPUT("Enter new password")

password1 = USERINPUT

OUTPUT("Enter password again")

password2 = USERINPUT

IF password1 = password2 THEN

match = True

OUTPUT ("Please continue")

ELSE

OUTPUT (Incorrect password – please re-enter a new password")

ENDIF

ENDWHILE

(see Python solution)

#password verification

match = False

while match == False:

print("Please enter new password: ")

password1 = input()

print("Please enter password again: ")

password2 = input()

if password1 == password2:

match = True

print("Please continue")

```
else:  
    print ("Incorrect password - please re-enter a new password: ")
```

- (b) Suggest two conditions for a new password that could be specified and validated.

For example:

"Password must be between 8 and 15 characters"

"Password must contain at least one numeric character"

Homework 3: Validation and Verification

Customers at a supermarket enter a monthly raffle to win a prize, which is the chance to receive the contents of their shopping cart free. To enter the raffle a customer must enter a number of random digits into a machine at the checkout, and the number is printed on their receipt.

The machine checks that the entry is valid using the pseudocode shown below. The function div performs integer division, so that for example $24 \text{ div } 7$ returns 3.

```
INPUT number  
n ← number  
d ← 0  
repeat  
    n ← n div 10  
    d ← d + 1  
UNTIL n = 0  
IF d <> 6 THEN  
    OUTPUT "Invalid entry"  
ELSE  
    OUTPUT "Entry accepted"
```

- (a) Put a tick in the correct column to show which message is displayed for each of the following entries. Use the trace table with columns headed n and d. [8]

number	"Entry accepted"	"Invalid entry"
--------	------------------	-----------------

103050	✓	
46729		✓
001854		✓
100000	✓	

Trace table		Trace table		Trace table		Trace table	
n	d	n	d	n	d	n	d
103050	0	46729	0	001854	0	100000	0
10305	1	4672	1	00185	1	10000	1
1030	2	467	2	0018	2	1000	2
103	3	46	3	001	3	100	3
10	4	4	4	00	4	10	4
1	5	0	5			1	5
0	6					0	6

(b) Formulate a rule for a valid raffle ticket number.

[2]

The number must be 6 digits long and must not have any leading zeroes
(See Python program)

```
#raffle ticket number
```

```
n = int(input("Enter 6 digit number, no leading zeros: "))
d = 0
while n>0:
    n = n//10
    d = d + 1
    print ("n,d", n,d)
if d!=6:
    print("Invalid entry")
else:
    print("Entry accepted")
```

2. A check digit for a five-digit product number is calculated as follows:

- assign weights of 6, 5, 4, 3, 2 to each of the digits of the number, starting from the most significant (leftmost) digit
- multiply each digit by its weight
- add the resulting numbers together
- divide by 10 using integer division
- Subtract the remainder from 10
- If the remainder is zero, the check digit is zero
- Otherwise, subtract the remainder from 10 and the result is the check digit

(a) Calculate the check digit for the number 25461.

[1]

Answer: 7

(b) The following pseudocode is designed to calculate the check digit of a 5-digit code as described in part (a) above.

The function MOD (a,b) finds the remainder when a is divided by b.

```
productCode = [0,0,0,0,0,0]
weight = 6
weightedSum = 0
OUTPUT ("Enter 5-digit product code without check digit")
FOR number = 1 to 5
    weightedDigit = productCode[number] * weight
    weightedSum = weightedSum + weightedDigit
    weight = weight - 1
ENDFOR
remainder = MOD (weightedSum, 10)
checkDigit = 10 - remainder <----- add new statement here
productCode[6] = checkDigit
```

There is a logic error in the program.

- (i) Complete the trace table below to find the value of the check digit and product code if the user enters the code 12345. [6]

productCode											
1	2	3	4	5	6	number	weighted Digit	weighted Sum	weight	remainder	check Digit
1	2	3	4	5				0	6		
						1	6	6	5		
						2	10	16	4		
						3	12	28	3		
						4	12	40	2		
						5	10	50	1	0	10

- (ii) Add one or more statements to correct the logic error. Where should the statement(s) be inserted? Indicate with an arrow on the pseudocode. [4]

The following statement should be inserted:

```

IF remainder = 0
    checkDigit = 0
ELSE
    checkDigit = 10 - remainder
ENDIF

```

Python code for Check Digit Program

```

#check digit program
#calculates check digit for a 5-character code
#to create a 6-digit product code
productCode = []
weight = 6
weightedSum = 0
productCode = input("Enter product code without check digit: ")

```

```

for number in range (5):

    print(productCode[number])
    weightedDigit = int(productCode[number]) * weight
    print("weighted digit = ",weightedDigit)
    weightedSum = weightedSum + weightedDigit
    weight = weight - 1
    print("weightedSum",weightedSum,"weight",weight)

#ENDFOR

remainder = weightedSum % 10
if remainder == 0:
    checkDigit = 0
else:
    checkDigit = 10 - remainder
#ENDIF

productCode = productCode + str(checkDigit)
print ("Product code = ",productCode)

```

Worksheet 4 Designing a database table

Worksheet is practical for this lesson

Homework 4 Designing a database table

A database table for a newsagent has the following fields:

Field name	Data type	Comment
PaperID	Short text , integer or Autonumber	
NewspaperName	Short text (or just "text")	
Price	Currency	
Frequency	Short text (or just "text")	Daily, weekly or monthly
SundayDelivery	Yes/No (or "Boolean")	Is it a Sunday paper?

- 2 (a) Suggest an appropriate data type for each of the fields. [5]
- (b) (i) Which field would you suggest should be the Primary key field? **PaperID** [1]
- (ii) What is special about a primary key field? [2]

It is unique – no two records can have the same value in this field. (Or, “It uniquely identifies the record.”)

Suggest two different validation rules that could be applied to fields in the database. [2]

Validation 1:	Field name:	Validation type:
	Price	Range check
Validation rule:		
Between 0.25 and 5.00 (or any reasonable range)		
Could be expressed as ≥ 0.25 AND ≤ 5.00		

Validation 2:	Field name:	Validation type:
	Frequency	Character check
Validation rule:		
Must be “D”, “W” or “M”		

Worksheet 5: Querying a database

Task 2: Using the logical operator OR in a query

1. The table tblFilm has the following records:

Film ID	Title	Studio	Release date	Production	Box Office(\$m)	Seen	Classification
1	Avatar	Fox	01 July 2009	254	2787.97	<input checked="" type="checkbox"/>	12
2	Spider-Man 3	Sony	16 April 2007	286	890.87	<input type="checkbox"/>	12
3	The Dark Knight Rises	WB	12 July 2012	230	1084.43	<input type="checkbox"/>	12
4	The Hobbit: The desolation of Smaug	WB	13 December 2013	225	960.37	<input checked="" type="checkbox"/>	12
5	Harry Potter and the half-blood Prince	WB	15 July 2009	268	934.42	<input type="checkbox"/>	U
6	Pirates of the Caribbean:Dead Man's Chest	BV	24 June 2006	256	1066.18	<input type="checkbox"/>	U
7	Shrek 2	DW	19 May 2004	100	919.83	<input checked="" type="checkbox"/>	U
8	Pirates of the Caribbean: At world's end	BV	19 May 2007	300	963.42	<input type="checkbox"/>	12
9	Skyfall	WB	23 October 2012	205	1108.56	<input type="checkbox"/>	12
10	Titanic	Fox	19 December 1997	260	2186.77	<input checked="" type="checkbox"/>	12

Write down the Film ID that will be found by the following query:

(Studio = “Fox”) OR (Studio = “BV”) 1, 6, 8, 10

2. Now create a new query in Access to check out the accuracy of your answer.

Tip: Your query grid should look something like this:

Field:	Film ID	Title	Studio	Release date	Production cost(\$m)	Box Office(\$m)	Classification
Table:	tblFilm	tblFilm	tblFilm	tblFilm	tblFilm	tblFilm	tblFilm
Sort:							
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			"Fox"				
or:			"BV"				

You can alternatively write the criteria on one line as: "Fox" OR "BV"

Save your query as qryStudio.

Close the database.

Task 3: Writing queries

Open the database Paintings1 and look at tblPainting.

Painting ID	Title	Artist	Price	Date sold	Price adjust
P06	Bal du moulin de la Galette	Pierre-Auguste Renoir	78,100,000	1990	£141,500,000
P05	Portrait of Dr. Gachet	Vincent van Gogh	82,500,000	1990	£149,400,000
P08	Woman III	Willem de Kooning	137,500,000	2006	£161,500,000
P03	The Card Players	Paul Cezanne	250,000,000	2011	£263,000,000
P07	Three studies of Lucian Freud	Francis Bacon	142,400,000	2013	£144,700,000
P04	Le Reve	Pablo Picasso	155,000,000	2013	£155,000,000
P02	Women of Algiers	Pablo Picasso	179,400,000	2015	£179,400,000
P01	Nafea Faa Ipoipo?	Paul Gauguin	300,000,000	2015	£300,000,000

(a) Write a query which will find all paintings sold before 2010 for more than \$100,000.

Date sold<2010 and price>100000

(b) Implement your query in Access. Which paintings were listed in the Results table?

(c) Which records will be displayed when the following query is run? **P04**

(Artist = "Pablo Picasso" OR Artist = "Paul Gauguin") AND (Price > 160,000,000)

(d) Implement the query in Access and check your answer is correct.

Homework 5: Querying a database

1. A secondhand car dealer keeps a database of cars for sale. A sample of the data is shown below.

CarRegistration	Make	Model	Year	Mileage	Net Selling Price	Transmissio	Doors
AV60HES	Peugot	1.0 Urban	2010	33156	£5,400.00	Manual	5
AX11UHY	Toyota	Aygo	2011	26875	£5,200.00	Manual	5
BH61LWZ	Hyundai	I10 1.2	2011	17411	£5,600.00	Manual	5
CF11YHK	Peugot	207	2011	35634	£5,100.00	Manual	3
EF11GTZ	Peugot	107	2011	50254	£4,880.00	Automatic	5
FD50GHK	Citroen	C4	2010	50766	£7,600.00	Automatic	3
GF52KLU	Renault	Megane	2012	31876	£5,900.00	Manual	3
GF59NGB	Renault	Clio	2009	18871	£5,400.00	Manual	3
GR12JUK	Vauxhall	Corsa	2012	40612	£6,500.00	Automatic	3

a) What is the purpose of a primary key in a table? **To uniquely identify the record** [1]

b) State which field would make a suitable primary key in this table. **CarRegistration** [1]

c) State the most appropriate data type for the following fields:

Mileage Integer [1]

Net selling price Currency [1]

d) State the registration number of the cars that will be found using each of the following criteria:

Mileage < 20000 BH61LWZ, GF59NGB [1]

Year >=2011 AND Transmission = "Automatic" GR12JUK [1]

(Make="Peugeot" OR "Renault") AND (Net Selling Price <= 5200.00) EF11GTZ [1]

e) Write the criteria for a query which will select all cars with 5 doors made in 2011 or after

Doors = 5 AND Year >= 2011 [3]

Python 3 cheatsheet (the basics)



Interact with the user (input and output)

Print a message

```
print('Hello, world!')
```

Print multiple values (of different types)

```
ndays = 365
print('There are', ndays, 'in a year')
```

Asking the user for a string

```
name = input('What is your name? ')
```

Asking the user for a whole number (an integer)

```
num = int(input('Enter a number: '))
```

Decide between options

Decide to run a block (or not)

```
x = 3
if x == 3:
    print('x is 3')
```

Are two values equal?

```
x == 3
```

△ two equals signs, not one

Are two values not equal?

```
x != 3
```

Less than another?

```
x < 3
```

Greater than another?

```
x > 3
```

Less than or equal to?

```
x <= 3
```

Greater than or equal to?

```
x >= 3
```

The answer is a Boolean:

```
True or False
```

elif can be used without else

elif can be used many times

```
mark = 80
if mark >= 50:
    print('credit')
elif mark >= 50:
    print('pass')
else:
    print('fail')
```

Decide between many blocks

```
mark = 80
if mark >= 65:
    print('credit')
elif mark >= 50:
    print('pass')
else:
    print('fail')
```

String manipulation

Compare two strings

```
msg = 'hello'
if msg == 'hello':
    print('howdy')
```

Convert to uppercase

```
msg.upper()
```

also lower and title

Count a character in a string

```
msg.count('l')
```

Replace a character or string

```
msg.replace('l', 'x')
```

Delete a character or string

```
msg.replace('l', '')
```

Is a character in a string?

```
'e' in msg
```

Is a string in another string?

```
'ell' in msg
```

also isupper and istitle

Text (strings)

Single quoted

```
'perfect'
```

Double quoted

```
"credit"
```

Multi-line

```
'''Hello;
World!'''
```

Add (concatenate) strings

```
'Hello' + 'World'
```

Multiply string by integer

```
'Echo...' * 4
```

Length of a string

```
len('Hello')
```

Convert string to integer

```
int('365')
```

Creating a variable

```
celsius = 25
```

Using a variable

```
celsius*9/5 + 32
```

Whole numbers (integers)

Addition and subtraction

```
365 + 1 - 2
```

Multiplication and division

```
25*9/5 + 32
```

Powers (2 to the power of 8)

```
2**8
```

Convert integer to string

```
str(365)
```

Repeat a block (a fixed number of times)

Count from 0 to 9

```
range(10)
```

△ range starts from 0 and goes up to, but not including, 10

Count from 1 to 10

```
range(1, 11)
```

Count from 10 down to 1

```
range(10, 0, -1)
```

Count 2 at a time to 10

```
range(0, 11, 2)
```

Count down 2 at a time

```
range(10, 0, -2)
```

Repeat a block over list (or string) indices

```
msg = 'I grok Python!'
for i in range(len(msg)):
    print(i, msg[i])
```

Putting it together: Celsius to Fahrenheit converter

Ask the user for a temperature in degrees Celsius

```
celsius = int(input('Temp. in Celsius: '))
```

Calculate the conversion

```
fahrenheit = celsius*9/5 + 32
```

Output the result

```
print(fahrenheit, 'Fahrenheit')
```



Relevant Quizlet Flashcards

Join my GCSE Team

<https://quizlet.com/join/rd4VKwq6F>

My GCSE CS Folder

<https://quizlet.com/myranT/folders/igcse-cs/sets>

Separate Quizlets

<https://quizlet.com/vn/549568759/igcse-cs-unit-07-algorithm-design-and-problem-solving-flash-cards/>

<https://quizlet.com/34502169/igcse-computerscience-databases-flash-cards/>