Summative Assessment 1 September – AS Level

MM:45 Time: 60 minutes

Name:…… Class:…….

1 (a) (i) Algorithms may be expressed using four basic constructs. One construct is sequence.

Complete the following table for two other constructs.

Construct	Pseudocode example
.....	....
.....	....

[4]

(ii) Simple algorithms usually consist of input, process and output.

Complete the table by placing ticks ('✓') in the relevant boxes.

Pseudocode statement	Input	Process	Output
`Temp ← SensorValue * Factor`			
`WRITEFILE "LogFile.txt", TextLine`			
`WRITEFILE "LogFile.txt", MyName & MyIDNumber`			
`READFILE "AddressBook.txt", NextLine`			

[4]

(b) Program variables have values as follows:

Variable	Value
Title	"101 tricks with spaghetti"
Version	'C'
Author	"Eric Peapod"
PackSize	4
WeightEach	6.2
Paperback	TRUE

(i) Evaluate each expression in the following table.
If an expression is invalid, write ERROR.

For the built-in functions list, refer to the **Appendix** on page 9

Expression	Evaluates to
MID(Title, 5, 3) & RIGHT(Author, 3)	
INT(WeightEach * PackSize)	
PackSize >= 4 AND WeightEach < 6.2	
LEFT(Author, ASC(Version) - 65)	
RIGHT(Title, (LENGTH(Author) - 6))	

[5]

(ii) Programming languages support different data types.

Give an appropriate data type for the following variables from **part (b)**.

Variable	Data type
Title	
Version	
PackSize	
WeightEach	
Paperback	

[5]

Q.2

Toni has a large collection of jazz CDs that are stored in different places. She wants to record where the CDs are stored. She decides to write a program to do this.

The program must store the data in a file, `MyMusic`.

(a) (i) Why is a file needed?

.....

.....[1]

(ii) `MyMusic` is a text file with the data for each CD as one line of text.

Data for a typical CD are:

Title: Kind of Green
Artist: Miles Coltrane
Location: Rack1-5

The line will be formed by concatenating the three data items.

For the example above, the line stored will be:

```
Kind of GreenMiles ColtraneRack1-5
```

Describe a problem that might occur when organising the data in this way.

.....

.....

.....

.....

Describe a possible solution.

.....

.....

.....

.....

[4]

(b) Toni must input the data into the file for all of her CDs.

A procedure, `InputData`, is needed to do this.

Toni designs the procedure and chooses the following identifiers:

Identifier	Data type
`CDTitle`	`STRING`
`CDArtist`	`STRING`
`CDLocation`	`STRING`

The procedure repeatedly performs the following steps:

- input a CD title (A rogue value of "##" is to be used to end the input)
- input the artist
- input the location
- create the text line
- write the text line to the file

When the rogue value is encountered the file is closed.

Write **program code** for the procedure `InputData`.

Visual Basic and Pascal: You should include declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language .....

```
def inputdata():
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[8]

Q.3 The data shown in **Figure** is a list of surnames of 20 motor car policyholders with the number of claims they have each made in the last five years.

	PolicyHolder
1	Wilcox
2	Adams
3	Pollard
4	Williams
5	Searle
6	Kelly
7	Lewis
8	Franks
9	Patel
10	Li Che
...	...
...	...
19	Wilkinson
20	Veale

	NoOfClaims
1	1
2	0
3	0
4	0
5	3
6	0
7	1
8	5
9	1
10	0
...	...
...	...
19	3
20	0

(a) Write pseudocode declaration statements for the PolicyHolder and NoOfClaims data structure above. [2]

1. .....

2. .....

(b) A new task is to design and write code to establish if there are any policyholders who have made five or more claims. The program will output a 'yes' or 'no' message only. Write the programcode for this task..
(Hint: Use a loop structure to initiate the loop, and then end the loop when some condition is met.)

.....
.....
.....
.....
.....
.....
.....
.....
.....

[6]

4 A program is to be written to calculate the discount given on purchases.

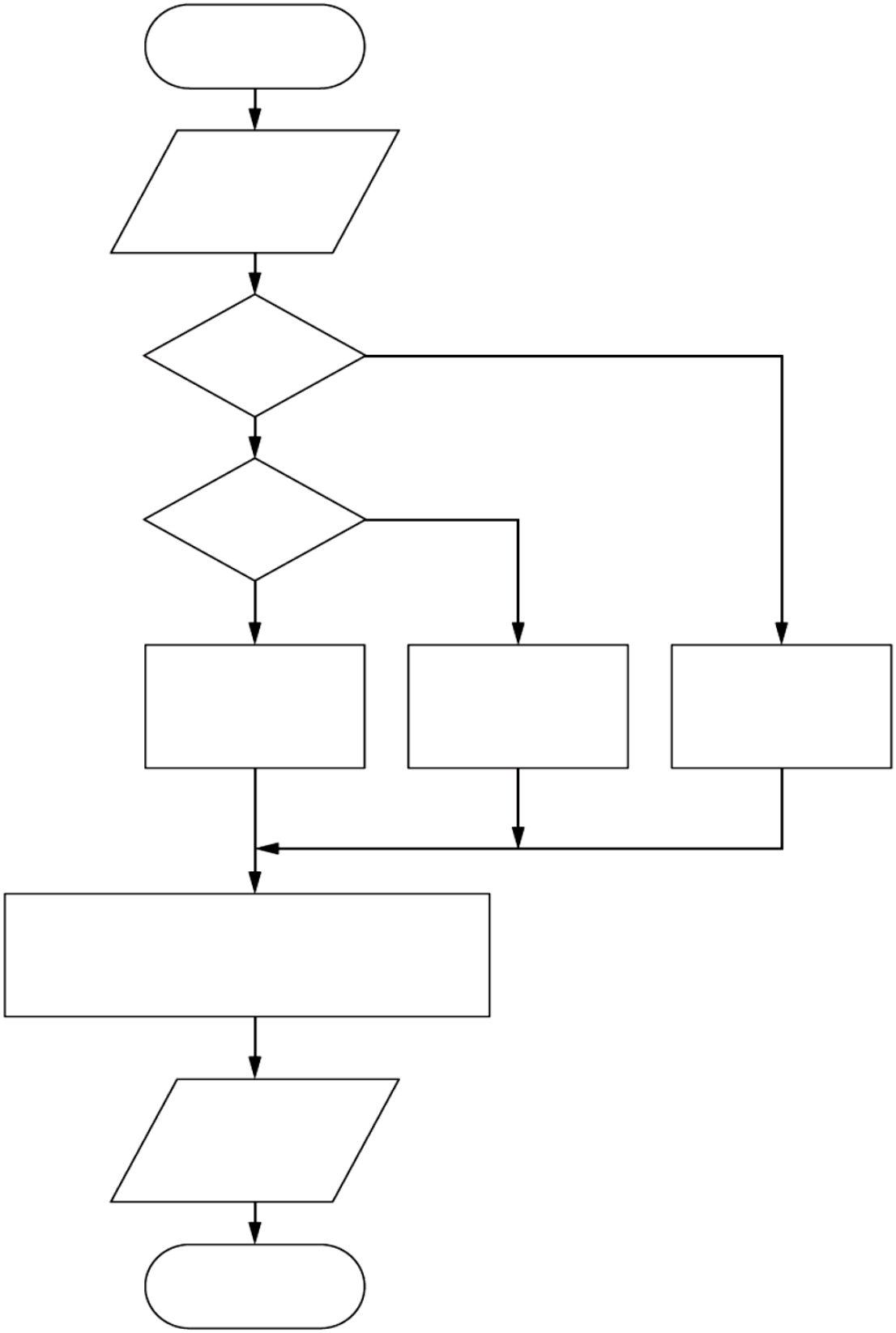
A purchase may qualify for a discount depending on the amount spent. The purchase price (`Purchase`), the discount rate (`DiscountRate`) and amount paid (`Paid`) is calculated as shown in the following pseudocode algorithm.

```
INPUT Purchase

IF Purchase > 1000
  THEN
    DiscountRate ← 0.10
  ELSE
    IF Purchase > 500
      THEN
        DiscountRate ← 0.05
      ELSE
        DiscountRate ← 0
    ENDIF
ENDIF

Paid ← Purchase * (1 - DiscountRate)
OUTPUT Paid
```

The algorithm is also to be documented with a program flowchart.

Complete the flowchart by:

- filling in the flowchart boxes
- labelling, where appropriate, lines of the flowchart



[6]

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `ThisString`

Example: `LENGTH("Happy Days")` returns `10`

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns `"ABC"`

`RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns `"FGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns the ASCII value of character `ThisChar`

Example: `ASC('A')` returns `65`

`RAND(x : INTEGER) RETURNS REAL`
returns a real number in the range `0` to `x` (`x` not inclusive).

Example: `RAND(87)` could return `35.43`

Operators (pseudocode)

Operator	Description
`&`	Concatenates (joins) two strings Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"`
`AND`	Performs a logical `AND` on two Boolean values Example: `TRUE AND FALSE` produces `FALSE`
`OR`	Performs a logical `OR` on two Boolean values Example: `TRUE OR FALSE` produces `TRUE`

Mark Scheme

1(a)(i) — **4**

Construct: Assignment
Pseudocode example: `Answer ← "YES"`

Construct: Selection
Pseudocode example: `IF X = 3 THEN OUTPUT "HELLO"`

Construct: Repetition / Iteration
Pseudocode example: `FOR N ← 1 to 100`

One mark for construct
One mark for pseudocode example
Maximum 4 marks

1(a)(ii) — **4**

Pseudocode statement	Input	Process	Output
`Temp ← SensorValue * Factor`		✓	
`WRITEFILE "LogFile.txt", TextLine`			✓
`WRITEFILE "LogFile.txt", MyName & MyIDNumber`		✓	✓
`READFILE "AddressBook.txt", NextLine`	✓	(✓)	

One mark per correct row

1(b)(i) — **5**

Expression	Evaluates to
`MID(Title, 5, 3) & RIGHT(Author, 3)`	**"tripod"**
`INT(WeightEach * PackSize)`	**24**
`PackSize >= 4 AND WeightEach < 6.2`	**FALSE**
`LEFT(Author, ASC(Version) - 65)`	**"Er"**
`RIGHT(Title, (LEN(Author) - 6))`	**"hetti"**

Quotes must be present
Must be capital E in row 4

1(b)(ii)

Variable	Data type
`Tile`	STRING
`Version`	CHAR
`PackSize`	INTEGER
`WeightEach`	REAL
`Paperback`	BOOLEAN

One mark per data type

5

Q.2

(a) (i)

- So that the data / information is saved after the program is run / when the computer is switched off
- So the data / information can be accessed next time the program is run
- So the data information can be "permanently stored"

Max 1

(ii)

Problem:

- When retrieving / searching for / editing (text relating to a particular CD)
- Can't tell where the artist name stops and the title begins (or any similar explanation or example)

Solution 1:

- Use of a separator character// or by example
- Where the separator character does not occur in the original strings

Solution 2:

- Use a fixed number of characters for each data item
- Data items are padded with e.g. <Space> character where needed

Solution 3:

- Convert original data items to CamelCase
- …and add a Space separator

Mark as follows:
Two marks for description of problem
Two marks for description of solution

4

(b) 'Pseudocode' solution included here for development and clarification of mark scheme.
Programming language solutions appear in the Appendix. **Max 8**

```
PROCEDURE InputData()

DECLARE CDTitle : STRING
DECLARE CDArtist : STRING
DECLARE CDLocation : STRING
DECLARE FileData : STRING


OPENFILE "MyMusic" FOR WRITE

OUTPUT "Input CD Title"
INPUT CDTitle

WHILE CDTitle <> "##"
      OUPUT "Input CD Artist"
      INPUT CDArtist
      OUPUT "Input CD Location"
      INPUT CDLocation
      FileData = CDTitle & ':' & CDArtist & ':' &
            CDLocation
      WRITEFILE "MyMusic.txt", FileData
      OUTPUT "Input CD Title"
      INPUT CDTitle
ENDWHILE

CLOSEFILE("MyMusic.txt ")

ENDPROCEDURE
```

One mark for each of the following:

- Procedure heading and ending
- Declaration of `CDTitle`, `CDArtist` and `CDLocation`
- Open file for writing (Allow `MyMusic` or `MyMusic.txt`)
- Working conditional loop structure including test for rogue value (including initial input of `CDTitle`)
- Input of three data values (`CDTitle`, `CDArtist` and `CDLocation`) **inside a loop**
- String concatenation of three variables **inside a loop**
- Write three variables in single line to file **inside a loop**
- Close file
- Use of string separator

Solutions may repeatedly `OPEN` – `WRITE` – `CLOSE` within the loop.In this case the first `OPEN` could be in `WRITE` or `APPEND` mode with all others in `APPEND`.

```
def InputData() :
      #CDTitle String (or CDTitle = "")
      #CDArtist String (or CDArtist = "")
      #CDLocation String (or CDLocation = "")

      FileHandle = open("MyMusic", "w")
      CDTitle = input("Input CD Title: ")
      while CDTitle != "##" :
            CDArtist = input("Input CD Artist: ")
            CDLocation = input("Input CD location: ")
            FileHandle.write(CDTitle + ":" + CDArtist + ":" + CDLocation)
            CDTitle = input("Input CD Title: ")
      FileHandle.close()
```

Q,3

All statements must have
correct identifier name
correct data type (String / Text // Integer / Byte / Word / Int / ShortInt / Short as appropriate) ;

In addition, either array must have
brackets to indicate an 'array'
19/20 to indicate a range ;
MAX 2

b)

(b)

Intialisation of counter or Boolean variable

P := 1 / P := 0 / For P := 1 to 20 // IsFound := False ;

Looping

LOOP UNTIL // DO WHILE // WHILE DO // REPEAT UNTIL and used at the beginning/end of a code block as appropriate ;

Some loop condition is met

(P = 20/21) OR IsFound = TRUE / P = 20/21 // IsFound = TRUE / IsFound ;

IF with use of the array

IF NoOfClaims[P] ;

Selection condition

>4 / >=5 ;

Loop counter incremented

P = P+1

Final output

Correct logic followed with OUTPUT 'Yes'

A. multiple times

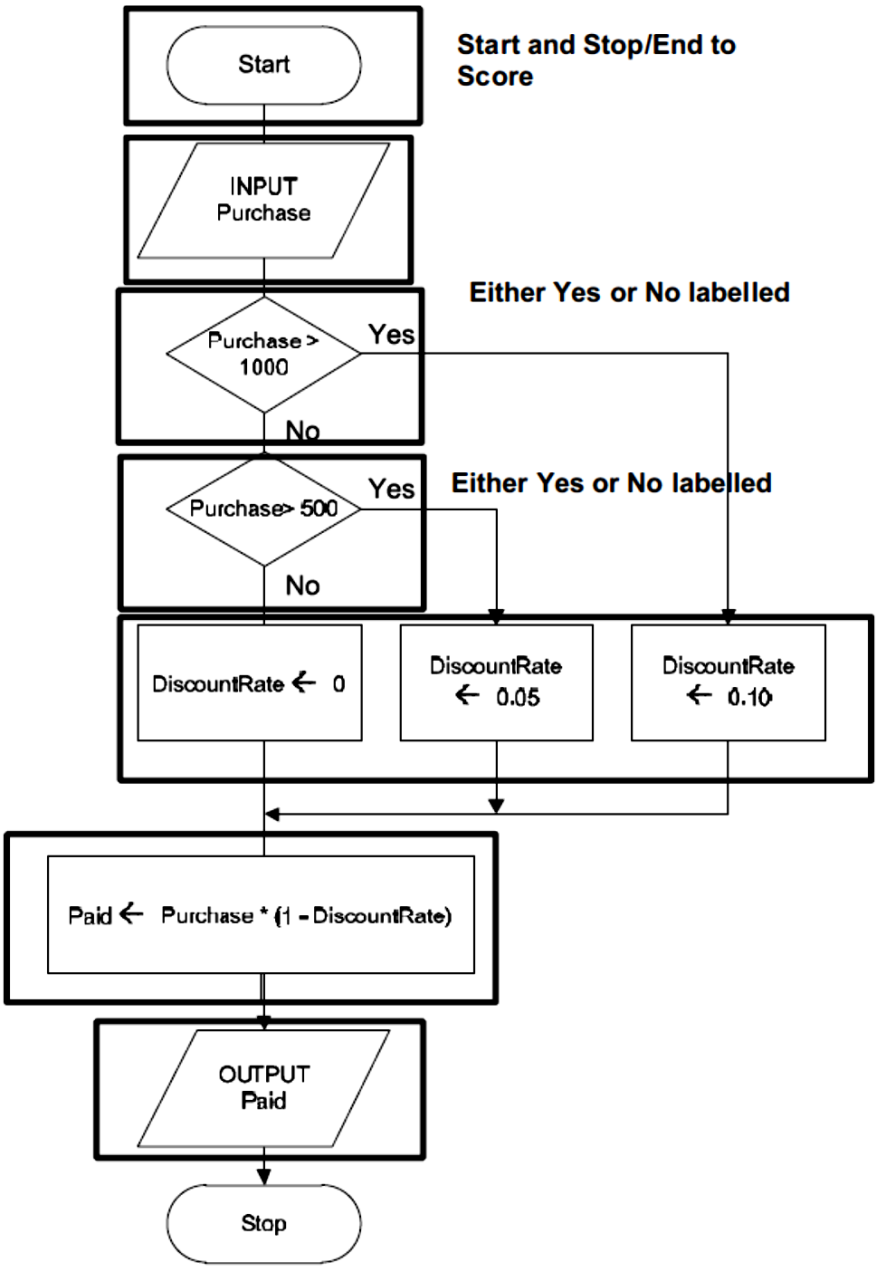
Final output

Correct logic followed with OUTPUT 'No'

R. Multiple times

R. 'Prose' scores 0

Q.4



Start and Stop/End to Score

Either Yes or No labelled

Either Yes or No labelled

[MAX 6]

[Total: 6]

[MAX 6]