QUESTION 2.

1 A marathon runner records their time for a race in hours, minutes and seconds.

An algorithm is shown below in structured English.

INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	Data type	Description
`RaceHours`	`INTEGER`	The hours part of the race time.

[3]

(b) Before the program is written, the design is amended.

The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged"
- "New personal best time"
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	Data type	Description

[1]

(ii) Write **program code** for the new design.

Visual Basic and Pascal: You should include the declaration statements for varia

Python: You should show a comment statement for each variable used with its data

Programming language .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[7]

(c) The program code will be tested using white-box testing.

(i) Explain what is meant by white-box testing.

.....

.....

.....[2]

(ii) Complete the table heading.

Complete Test Number 1.

Add the data for Test Number 2 and Test Number 3.

	Input values				Output	
Test number	**Race hours**	**Race minutes**	**Race seconds**	**.....**	**Total time (seconds)**	**Message**
1	3	4	13	11053	11053	
2				11053		
3				11053		

[6]

QUESTION 3.

6 A multi-user computer system makes use of passwords.

To be valid, a password must comply with the following rules:

- at least two lower-case alphabetic characters
- at least two upper-case alphabetic characters
- at least three numeric characters
- alpha-numeric characters only

A function, `ValidatePassword`, is needed to check that a given password follows these rules. This function takes a string, `Pass`, as a parameter and returns a Boolean value:

- `TRUE` if `Pass` contains a valid password
- `FALSE` otherwise.

(a) Write **program code** to implement the new function `ValidatePassword`.

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[10]

(b) (i) The function will be tested.

Give a valid string to check that the function returns `TRUE` under the correct c

String1: .....

Modify the valid string given for String1 to test each rule separately.

Explain your choice in each case.

String2: .....

Explanation: .....

.....

.....

String3: .....

Explanation: .....

.....

.....

String4: .....

Explanation: .....

.....

.....

String5: .....

Explanation: .....

.....

.....

[5]

(ii) When testing a module, it is necessary to test all possible paths through the code.

State the name given to this type of testing.

.....[1]

(iii) A program consisting of several modules may be tested using a pr
stub testing.

Explain this process.

.....

.....

.....

.....[2]

Appendix

Built-in functions (pseudocode)

In each function, if the function call is not properly formed, the function returns an error.

```
MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING
```

returns string of length `y` starting at position `x` from `ThisString`.
Example: **`MID("ABCDEFGH", 2, 3)`** returns string **`"BCD"`**

```
LENGTH(ThisString : STRING) RETURNS INTEGER
```

returns the integer value representing the length of string `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 string **`"ABC"`**

```
RIGHT(ThisString: STRING, x : INTEGER) RETURNS STRING
```

returns rightmost `x` characters from `ThisString`.
Example: **`RIGHT("ABCDEFGH", 3)`** returns string **`"FGH"`**

```
LCASE(ThisChar : CHAR) RETURNS CHAR
```

returns the character value representing the lower case equivalent of `ThisChar`.
If `ThisChar` is not an upper-case alphabetic character then it is returned unchanged.
Example: **`LCASE('W')`** returns **`'w'`**

```
MOD(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER
```

returns the integer value representing the remainder when `ThisNum` is divided by `ThisDiv`.
Example: **`MOD(10,3)`** returns **`1`**

```
DIV(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER
```

returns the integer value representing the whole number part of the result when `ThisNum` is divided by `ThisDiv`.
Example: **`DIV(10,3)`** returns **`3`**

Operators (pseudocode)

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

BLANK PAGE

QUESTION 4.

3 **(a)** State why a high-level language program must be translated before it can be

.....

.....

(b) A program runs but does not give the expected output.

Describe **two** methods you could use to find the error.

Method 1 .....

.....

.....

.....

Method 2 .....

.....

.....

.....

[4]

(c) Two testing methods are black-box and white-box. A student is choosing test data for both methods.

Tick **one or more** boxes in each row to identify the testing method each statement describes.

Statement	White-box	Black-box
The student does not need to know the structure of the code.		
The student chooses data to test every possible path through the code.		
The student chooses normal, boundary and erroneous data.		
The student chooses data to test that the program meets the specification.		

[4]

Question 4 begins on the next page.

QUESTION 5.

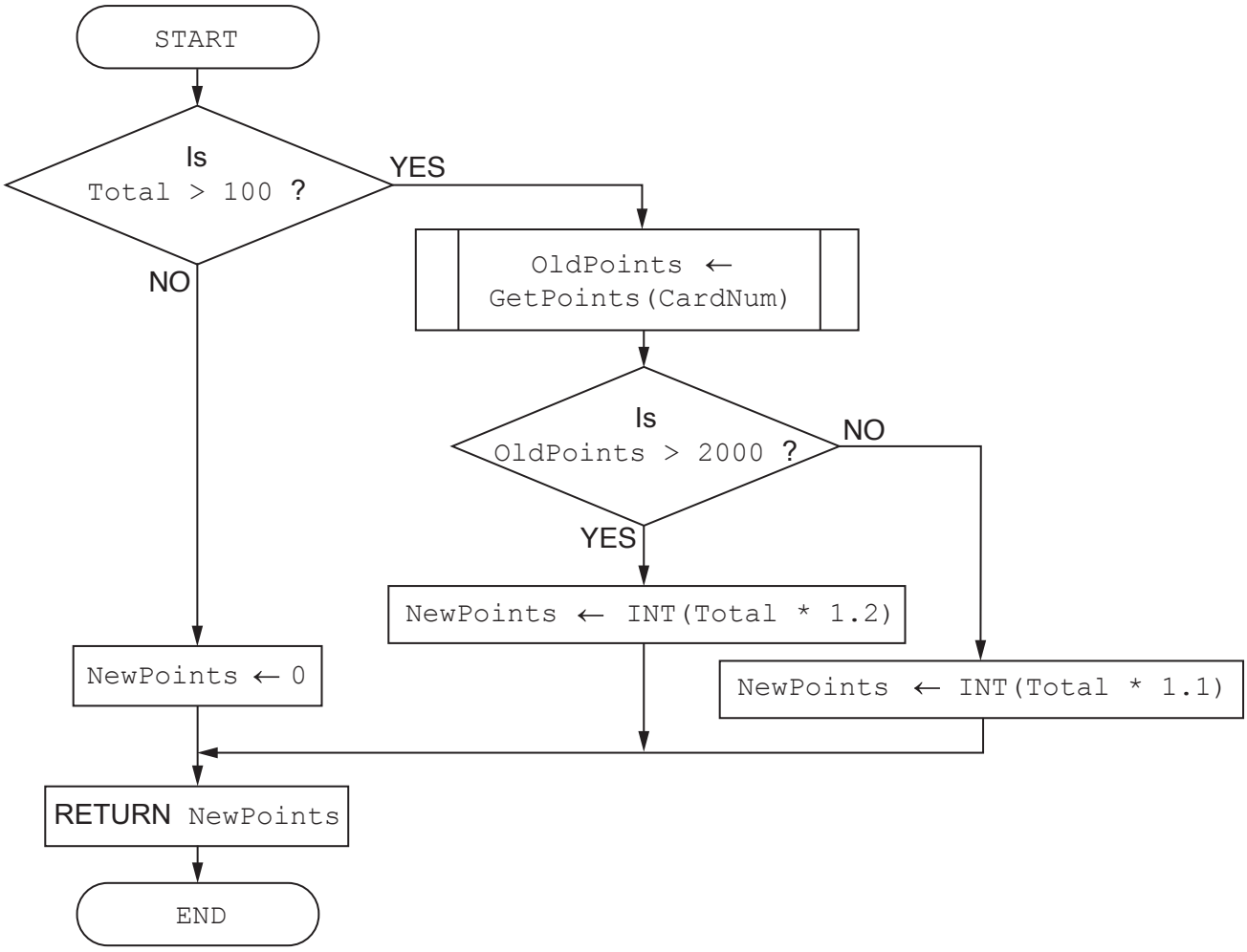
2 Shop customers have a discount card with a unique card number. Customers co
time they buy items. The number of points they collect depends on:

- the total amount they spend
- the number of points already collected.

The function `CalcPoints()` takes the card number and the total amount spent as parameters. I
returns the number of new points collected. A flowchart for the function is shown.

The function uses the following variables and functions.

Identifier	Data type	Description
`CardNum`	`STRING`	A numeric string representing the unique card number
`OldPoints`	`INTEGER`	The number of points already collected
`NewPoints`	`INTEGER`	The number of new points collected
`Total`	`REAL`	The amount spent
`GetPoints()`	`FUNCTION`	Takes the card number as a parameter and returns the number of points already collected
`INT()`	`FUNCTION`	Refer to the **Appendix** on page 16



(a) (i) Write **pseudocode** for the `CalcPoints()` function.

Your solution should follow the flowchart as closely as possible.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[7]

(ii) The value of the total amount spent is calculated by an Electronic Point
system. This system does not have the prices of all items. For these item
amount has to be entered manually.

A function, `GetTotal()`, prompts the user to input this value.

If the user enters a valid value greater than 0 and less than 10 000, the function returns the value. The function prompts the user to re-enter the value each time the user enters an invalid value.

Write **pseudocode** to complete the `GetTotal()` function.

```
FUNCTION GetTotal() RETURNS REAL
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
ENDFUNCTION
```

[5]

(b) The function `CalcPoints()` is written in a high-level language. It has bee
does not contain any syntax or logic errors.

(i) Name **and** describe **one** other type of error that the high-level language co
contain.

Name .....

Description .....

.....

.....

[2]

(ii) The function `CalcPoints()` is tested using white-box testing.

State **two** different values of `Total` that could be used to test different paths through the algorithm. Justify your choices.

Value .....

Justification .....

.....

.....

Value .....

Justification .....

.....

.....

[4]

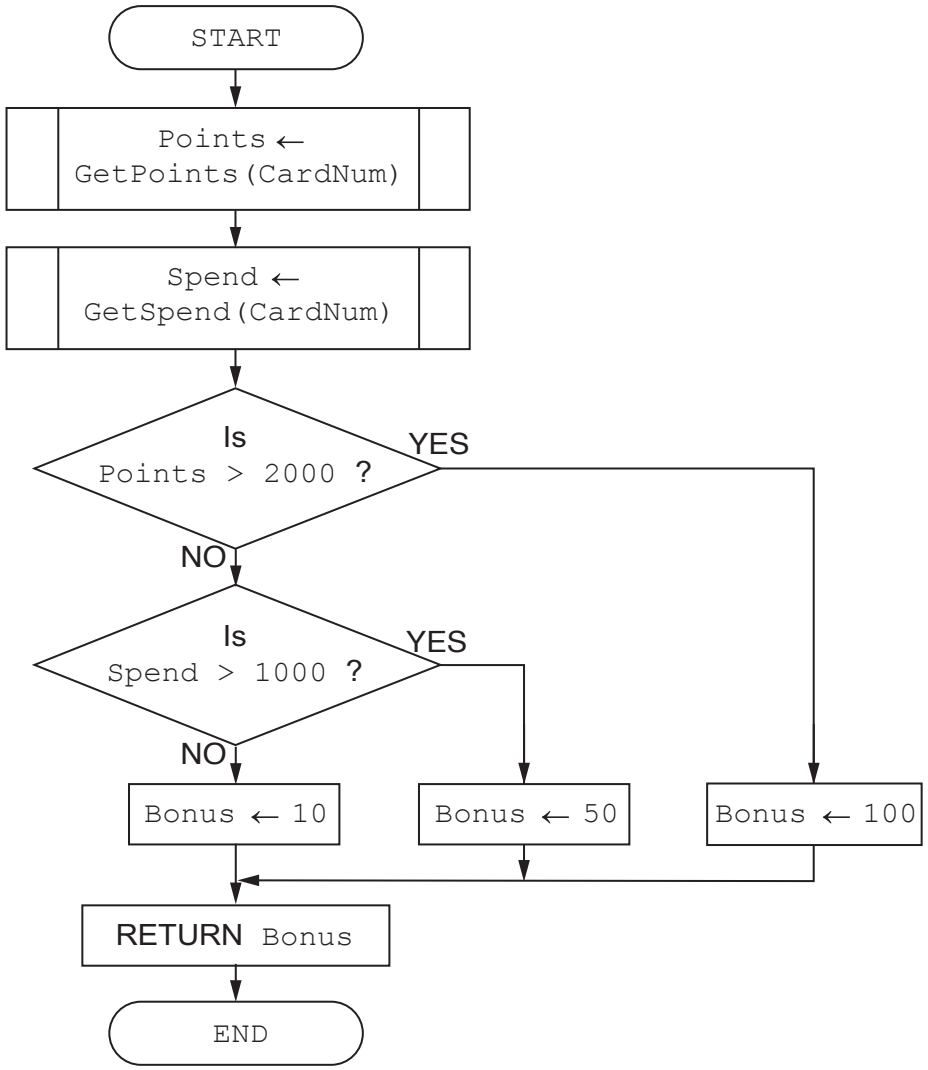
QUESTION 6.

2 Shop customers have a discount card with a unique card number. Customers co
they buy items. At the end of each year, customers are given bonus (extra) points
total amount they have spent during the year, and the number of points they have on t

The function `CalcBonus()` takes the card number as a parameter. It returns the bonus poi
the customer. A flowchart for the function is shown.

The function uses the following variables and functions.

Identifier	Data type	Description
`CardNum`	`STRING`	A numeric string representing the unique card number
`Points`	`INTEGER`	The number of points collected
`Spend`	`REAL`	The total amount that customer has spent during the year
`Bonus`	`INTEGER`	The number of bonus points
`GetPoints()`	`FUNCTION`	Takes the card number as a parameter and returns the number of points already collected
`GetSpend()`	`FUNCTION`	Takes the card number as a parameter and returns the total amount that customer has spent during the year



(a) (i) Write **pseudocode** for the `CalcBonus()` function.

Your solution should follow the flowchart for the function as closely as possib

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

(ii) The function `GetCardNumber()` prompts the user to input a card n
number input is valid.

A valid card number has 16 characters. Each character is a numeric characte
`'9'`).

Write **pseudocode** to complete the `GetCardNumber()` function.

You should refer to the function `IS_NUM()` in the **Appendix** on page 16.

```
FUNCTION GetCardNumber() RETURNS STRING
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
ENDFUNCTION
```

[6]

(b) The function `CalcBonus()` is written in a high-level language.

(i) The function is tested using black-box testing and does not contain any syntax errors.

Name **and** describe **one** other type of error that black-box testing could find.

Name .....

Description .....

.....

.....

[2]

(ii) The function `CalcBonus()` is tested using white-box testing.

State **two** different pairs of values for `Spend` and `Points` that can be used to test different paths through the function. Justify your choices.

`Spend` ..... `Points` .....

Justification .....

.....

.....

`Spend` ..... `Points` .....

Justification .....

.....

.....

[4]

(c) Name **two** types of program maintenance **and** state the reason why each is needed.

Name .....

Reason .....

.....

.....

Name .....

Reason .....

.....

.....

[4]

QUESTION 7.

1 **(a)** **(i)** Algorithms may be expressed using four basic constructs. One construc

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 16.

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]

(c) White-box and black-box are two types of testing. In white-box testing, data are chosen to test every possible path through the program.

Explain how data are chosen in black-box testing.

.....

..... [2]

QUESTION 8.

5 Nigel is learning about string handling. He wants to write code to count the numb
given string. A word is defined as a sequence of alphabetic characters that is separa
more space characters.

His first attempt at writing an algorithm in pseudocode is as follows:

```
PROCEDURE CountWords(Message : STRING)

   DECLARE NumWords : INTEGER
   DECLARE Index : INTEGER
   CONSTANT Space = ' '

   NumWords ← 0

   FOR Index ← 1 TO LENGTH(Message)
      IF MID(Message, Index, 1) = Space
         THEN
            NumWords ← NumWords + 1
      ENDIF
   ENDFOR

   OUTPUT "Number of words : " , NumWords

ENDPROCEDURE
```

For the built-in functions list, refer to the **Appendix** on page 18.

His first attempt is incorrect. He will use white-box testing to help him to identify the problem.

(a) (i) State the purpose of white-box testing.

.....

..... [1]

(ii) Dry running the code is often used in white-box testing. In this method, the programmer records the values of variables as they change.

Identify what the programmer would normally use to record the changes.

..... [1]

(b) (i) Write a test string containing two words that gives the output:

```
Number of words : 2
```

Use the symbol `'∇'` to represent each space character in your test string.

Explain why the algorithm gives the output shown above.

String .....

Explanation .....

.....

.....

.....

.....

[3]

(ii) Nigel tested the procedure with the strings:

String 1: `"Red∇and∇Yellow"`
String 2: `"Green∇∇and∇∇Pink∇"`

Give the output that is produced for each of the strings.

Describe the changes that would need to be made to the algorithm to give the correct output in each case.

Do **not** write pseudocode **or** program code.

String 1 .....

Description .....

.....

.....

.....

String 2 .....

Description .....

.....

.....

.....

[6]

QUESTION 9.

4 A program is being written to control the operation of a portable music player. The program controls the output volume.

The player has two buttons, one to increase the volume and one to decrease it. When a button is pressed, a procedure `Button()` is called with a parameter value representing the button as follows:

Button	Parameter value
Volume increase	10
Volume decrease	20

For example, pressing the volume increase button three times followed by pressing the volume decrease button once would result in the calls:

```
CALL Button(10) // VolLevel increased by 1
CALL Button(10) // VolLevel increased by 1
CALL Button(10) // VolLevel increased by 1
CALL Button(20) // VolLevel decreased by 1
```

The program makes use of two global variables of type `INTEGER` as follows:

Variable	Description
`VolLevel`	The current volume setting. This must be in the range 0 to 49.
`MaxVol`	A value that can be set to limit the maximum value of `VolLevel`, in order to protect the user's hearing. A value in the range 1 to 49 indicates the volume limit. A value of zero indicates that no volume limit has been set.

The procedure `Button()` will modify the value of `VolLevel` depending on which button has been pressed and whether a maximum value has been set.

(a) Write **pseudocode** for the procedure `Button()`. Declare any additional var

The value of `MaxVol` should **not** be changed within the procedure.

Parameter validation is **not** necessary.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(b) The procedure `Button()` is to be tested using black-box testing.

Fill in the gaps below to define **three** tests that could be carried out.

TEST 1 – `VolLevel` is changed

Parameter value: 10

`MaxVol`: .....

`VolLevel` value before call to `Button()`: 48

`VolLevel` expected value after call to `Button()`: .....

TEST 2 – `VolLevel` is **not** changed

Parameter value: 10

`MaxVol`: 34

`VolLevel` value before call to `Button()`: .....

`VolLevel` expected value after call to `Button()`: .....

TEST 3 – `VolLevel` is **not** changed

Parameter value: .....

`MaxVol`: 40

`VolLevel` value before call to `Button()`: 0

`VolLevel` expected value after call to `Button()`: .....

[6]

(c) The testing stage is part of the program development cycle.

(i) The program for the music player has been completed. The program does not have any syntax errors, but testing could reveal further errors.

Identify **and** describe **one different** type of error that testing could reveal.

Type .....

Description .....

.....

.....

[2]

(ii) Stub testing is a technique often used in the development of modular programs.

Describe the technique.

.....

.....

.....

.....

.....

..... [3]