QUESTION 8.

5 A company creates two new websites, Site X and Site Y, for selling bicycles.

Various programs are to be written to process the sales data.

These programs will use data about daily sales made from Site X (using variable `SalesX`) and Site Y (using variable `SalesY`).

Data for the first 28 days is shown below.

	SalesDate	SalesX	SalesY
1	03/06/2015	0	1
2	04/06/2015	1	2
3	05/06/2015	3	8
4	06/06/2015	0	0
5	07/06/2015	4	6
6	08/06/2015	4	4
7	09/06/2015	5	9
8	10/06/2015	11	9
9	11/06/2015	4	1
…			
28	01/07/2015	14	8

(a) Name the data structure to be used in a program for `SalesX`.

.....[2]

(b) The programmer writes a program from the following pseudocode design.

```
x ← 0
FOR DayNumber ← 1 TO 7
   IF SalesX[DayNumber] + SalesY[DayNumber] >= 10
      THEN
         x ← x + 1
         OUTPUT SalesDate[DayNumber]
   ENDIF
ENDFOR
OUTPUT x
```

(i) Trace the execution of this pseudocode by completing the trace table below.

x	DayNumber	OUTPUT
0		

[4]

(ii) Describe, in detail, what this algorithm does.

.....

.....

.....

.....[3]

(c) The company wants a program to output the total monthly sales for one websites.

The programmer codes a function with the following function header:

```
FUNCTION MonthlyWebSiteSales(ThisMonth : INTEGER, ThisSite : CHAR)
                                                    RETURNS INTEGER
```

The function returns the total number of bicycles sold for the given month and website.

The function will use the following:

Identifier	Data type	Description
`ThisMonth`	`INTEGER`	Represents the month number e.g. 4 represents April
`ThisSite`	`CHAR`	Coded as: • X for website X • Y for Website Y

(i) Give the number of parameters of this function. .....[1]

(ii) Some of the following function calls may be invalid.

Mark each call with:

- a tick (✓), for a valid call
- a cross (✗), for an invalid call

For any function calls which are invalid, explain why.

Function call	Tick (✓) / cross (✗)	Explanation (if invalid)
`MonthlyWebSiteSales(1, "Y")`		
`MonthlyWebSiteSales(11, 'X', 'Y')`		
`MonthlyWebSiteSales(12, 'X')`		

[3]

(d) The company decides to offer a discount on selected dates. A program is w
the dates on which a discount is offered.

The program creates a text file, DISCOUNT_DATES (with data as shown), for a nu
consecutive dates.

03/06/2015 TRUE
04/06/2015 FALSE
05/06/2015 FALSE
06/06/2015 FALSE
07/06/2015 FALSE
08/06/2015 FALSE
09/06/2015 FALSE
10/06/2015 TRUE
11/06/2015 FALSE

01/07/2015 FALSE

Each date and discount indicator is separated by a single <Space> character.

The discount indicators are:

- FALSE – indicates a date on which no discount is offered
- TRUE – indicates a date on which a discount is offered

A programming language has the built-in function `CONCAT` defined as follows:

```
CONCAT(String1 : STRING, String2 : STRING [, String3 : STRING] )
                                                    RETURNS STRING
```

For example: `CONCAT("San", "Francisco") returns "SanFrancisco"`
`CONCAT("New", "York", "City") returns "NewYorkCity"`

The use of the square brackets indicates that the parameter is optional.

The following incomplete pseudocode creates the text file DISCOUNT_DATES.

Complete the pseudocode.

```
OPENFILE "DISCOUNT_DATES" FOR .....
INPUT .....
WHILE  NextDate <>"XXX"
   INPUT Discount
   ..... = CONCAT(NextDate, " ", Discount)
   WRITEFILE "DISCOUNT_DATES",  NextLine
   INPUT NextDate
.....
OUTPUT "File now created"
CLOSEFILE
```

[4]

Question 5(e) continues on page 18.

(e) The `DISCOUNT_DATES` text file is successfully created.

The company now wants a program to:

- key in a date entered by the user
- search the text file for this date
- if found, output one of the following messages:
    - "No discount on this date"
    - "This is a discount date"
- if not found, output "Date not found"

(i) Add to the identifier table to show the variables you need for this new program.

Identifier	Data type	Description
`DISCOUNT_DATES`	`FILE`	Text file to be used

[3]

(ii) Write the **program code**.

Do not include any declaration or comment statements for the variables used.

Programming language .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[7]

QUESTION 9.

5 A firm employs workers who assemble amplifiers. Each member of staff works a... of hours each day.

The firm records the number of completed amplifiers made by each employee each day.

Management monitor the performance of all its workers.

Production data was collected for 3 workers over 4 days.

	Daily hours worked
Worker 1	5
Worker 2	10
Worker 3	10

Production data

	Worker 1	**Worker 2**	**Worker 3**
Day 1	10	20	9
Day 2	11	16	11
Day 3	10	24	13
Day 4	14	20	17

A program is to be written to process the production data.

(a) The production data is to be stored in a 2-dimensional array `ProductionData`, declared as follows:

```
DECLARE ProductionData ARRAY[1:4, 1:3] : INTEGER
```

(i) Describe **two** features of an array.

1 .....

.....

2 .....

.....[2]

(ii) Give the value of `ProductionData[3, 2]`.

.....[1]

(iii) Describe the information produced by the expression:

```
ProductionData[2, 1] + ProductionData[2, 2] + ProductionData[2, 3]
```

.....

.....[2]

(b) Complete the trace table for the pseudocode algorithm below.

```
FOR WorkerNum ← 1 TO 3
   WorkerTotal[WorkerNum] ← 0
ENDFOR

FOR WorkerNum ← 1 TO 3
   FOR DayNum ← 1 TO 4
      WorkerTotal[WorkerNum] ← WorkerTotal[WorkerNum] +
                                    ProductionData[DayNum, WorkerNum]
   ENDFOR
ENDFOR

FOR WorkerNum ← 1 TO 3
   WorkerAverage ← WorkerTotal[WorkerNum]/
                                    (4 * DailyHoursWorked[WorkerNum])
   IF WorkerAverage < 2
      THEN
         OUTPUT "Investigate", WorkerNum
   ENDIF
ENDFOR
```

WorkerNum	DayNum	WorkerAverage	OUTPUT

WorkerTotal		
1	2	3

[8]

(c) An experienced programmer suggests that the pseudocode would be best in procedure `AnalyseProductionData`.

Assume that both arrays, `DailyHoursWorked` and `ProductionData`, are availab procedure from the main program and they are of the appropriate size.

```
PROCEDURE AnalyseProductionData(NumDays : INTEGER, NumWorkers : INTEGER)

   DECLARE .....

   DECLARE .....

   DECLARE .....

   DECLARE .....

   FOR WorkerNum ← 1 TO 3
      WorkerTotal[WorkerNum] ← 0
   ENDFOR

   FOR WorkerNum ← 1 TO 3
      FOR DayNum ← 1 TO 4
         WorkerTotal[WorkerNum] ← WorkerTotal[WorkerNum] +
                                        ProductionData[DayNum, WorkerNum]
      ENDFOR
   ENDFOR

   FOR WorkerNum ← 1 TO 3
      WorkerAverage ← WorkerTotal[WorkerNum]/
                                       (4 * DailyHoursWorked [WorkerNum])
      IF WorkerAverage < 2
         THEN
            OUTPUT "Investigate", WorkerNum
      ENDIF
   ENDFOR

ENDPROCEDURE
```

(i) Complete the declaration statements showing the local variables. [4]

(ii) The original pseudocode has been 'pasted' under the procedure header.

Circle all the places in the original pseudocode where changes will need to be made.

Write the changes which need to be made next to each circle. [3]

(iii) Write the statement for a procedure call which processes data for 7 days for 13 workers.

.....[1]

BLANK PAGE

16

BLANK PAGE

QUESTION 10.

5 A program is to process a set of integers.

The integers are stored in an array, `Num`. The first `N` elements are to be processed.

The pseudocode for this program is shown below:

```
FOR i ← 1 TO (N - 1)
   j ← 1
   REPEAT
      IF Num[j] > Num[j + 1]
         THEN
            Temp ← Num[j]
            Num[j] ← Num[j + 1]
            Num[j + 1] ← Temp
      ENDIF
      j ← j + 1
   UNTIL j = (N - i + 1)
ENDFOR
```

(a) (i) Trace the execution of the pseudocode for the value `N` = 5 and the given array of integers.

N	i	j	Temp		Num 1	2	3	4	5
5					11	16	13	7	8

[8]

(ii) State the purpose of the algorithm.

.....

.....

(iii) Describe what evidence from the trace table suggests that the given pseudocode is inefficient.

.....

.....[1]

(b) Complete the identifier table documenting the use of each of the variables.

Identifier	Data type	Description
`Num`	`ARRAY[1:100] OF INTEGER`	The array of numbers.
`N`		
`i`		
`j`		
`Temp`		

[5]

QUESTION 11.

6 A string-handling function has been developed. The pseudocode for this function

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

```
FUNCTION SSM(String1, String2 : STRING) RETURNS INTEGER
    DECLARE n, f, x, y : INTEGER

    n ← 0
    f ← 0

    REPEAT
        n ← n + 1
        x ← n
        y ← 1
        WHILE MID(String1, x, 1) = MID(String2, y, 1)

            IF y = LENGTH(String2)
                THEN
                    f ← n
                ELSE
                    x ← x + 1
                    y ← y + 1
            ENDIF

        ENDWHILE

    UNTIL (n = LENGTH(String1)) OR (f <> 0)

    RETURN f

ENDFUNCTION
```

(a) Complete the trace table below by performing a dry run of the function when it is called as follows:

```
SSM("RETRACE", "RAC")
```

n	f	x	y	MID(String1, x, 1)	MID(String2, y, 1)
0	0				

[6]

(b) (i) Describe the purpose of function `SSM`.

.....

.....

.....

.....[2]

(ii) One of the possible return values from function `SSM` has a special meaning.

State the value and its meaning.

Value .....

Meaning .....
[2]

(iii) There is a problem with the logic of the pseudocode. This could generate a run-time error.

Describe the problem.

.....

.....

.....

.....[2]

Appendix

Built-in functions

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

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

returns the string of length `y` starting at position `x` from `ThisString`

Example: **`MID("ABCDEFGH", 2, 3)`** will return string **`"BCD"`**

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

returns the leftmost `x` characters from `ThisString`

Example: **`LEFT("ABCDEFGH", 3)`** will return string **`"ABC"`**

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

returns the rightmost `x` characters from `ThisString`

Example: **`RIGHT("ABCDEFGH", 3)`** will return string **`"FGH"`**

`ASC(ThisChar : CHAR) RETURNS INTEGER`

returns the ASCII value of character `ThisChar`

Example: **`ASC('W')`** will return **`87`**

`LENGTH(ThisString : STRING) RETURNS INTEGER`

returns the integer value representing the length of string `ThisString`

Example: **`LENGTH("Happy Days")`** will return **`10`**

String operator

`&` operator

concatenates (joins) two strings

Example: **`"Summer" & " " & "Pudding"`** produces **`"Summer Pudding"`**

BLANK PAGE

QUESTION 12.

6 A string-handling function has been developed. The pseudocode for this function

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

```
FUNCTION SSM(String1, String2 : STRING) RETURNS INTEGER
    DECLARE n, f, x, y : INTEGER

    n ← 0
    f ← 0

    REPEAT
        n ← n + 1
        x ← n
        y ← 1
        WHILE MID(String1, x, 1) = MID(String2, y, 1)

            IF y = LENGTH(String2)
                THEN
                    f ← n
                ELSE
                    x ← x + 1
                    y ← y + 1
            ENDIF

        ENDWHILE

    UNTIL (n = LENGTH(String1)) OR (f <> 0)

    RETURN f

ENDFUNCTION
```

(a) Complete the trace table below by performing a dry run of the function when it is called as follows:

```
SSM("RETRACE", "RAC")
```

n	f	x	y	MID(String1, x, 1)	MID(String2, y, 1)
0	0				

[6]

(b) (i) Describe the purpose of function `SSM`.

.....

.....

.....

.....[2]

(ii) One of the possible return values from function `SSM` has a special meaning.

State the value and its meaning.

Value .....

Meaning .....
[2]

(iii) There is a problem with the logic of the pseudocode. This could generate a run-time error.

Describe the problem.

.....

.....

.....

.....[2]

Appendix

Built-in functions

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

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

returns the string of length `y` starting at position `x` from `ThisString`

Example: **`MID("ABCDEFGH", 2, 3)`** will return string **`"BCD"`**

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

returns the leftmost `x` characters from `ThisString`

Example: **`LEFT("ABCDEFGH", 3)`** will return string **`"ABC"`**

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

returns the rightmost `x` characters from `ThisString`

Example: **`RIGHT("ABCDEFGH", 3)`** will return string **`"FGH"`**

`ASC(ThisChar : CHAR) RETURNS INTEGER`

returns the ASCII value of character `ThisChar`

Example: **`ASC('W')`** will return **`87`**

`LENGTH(ThisString : STRING) RETURNS INTEGER`

returns the integer value representing the length of string `ThisString`

Example: **`LENGTH("Happy Days")`** will return **`10`**

String operator

`&` operator

concatenates (joins) two strings

Example: **`"Summer" & " " & "Pudding"`** produces **`"Summer Pudding"`**

BLANK PAGE

QUESTION 13.

6 A string-handling function has been developed.

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

The pseudocode for this function is shown below.

```
FUNCTION SF(ThisString : STRING) RETURNS STRING
   DECLARE x          : CHAR
   DECLARE NewString  : STRING
   DECLARE Flag       : BOOLEAN
   DECLARE m, n       : INTEGER

   Flag ← TRUE
   NewString ← ""
   m ← LENGTH(ThisString)

   FOR n ← 1 TO m

      IF Flag = TRUE
         THEN
            x ← UCASE(MID(ThisString, n, 1))
            Flag ← FALSE
         ELSE
            x ← LCASE(MID(ThisString, n, 1))
      ENDIF

      NewString ← NewString & x

      IF x = " "
         THEN
            Flag ← TRUE
      ENDIF

   ENDFOR

   RETURN NewString
ENDFUNCTION
```

(a) (i) Complete the trace table below by performing a dry run of the function when it is called as follows:

```
SF("big BEN")
```

n	x	Flag	m	NewString

[4]

(ii) Describe the purpose of function `SF`.

.....

.....

.....

.....[2]

(b) Test data must be designed for the function `SF`.

(i) State what happens when the function is called with an empty string.

.....

.....[1]

(ii) The function should be thoroughly tested.

Give **three** examples of non-empty strings that may be used.

In each case explain why the test string has been chosen.

String .....

Explanation .....

.....

String .....

Explanation .....

.....

String .....

Explanation .....

.....

[3]

QUESTION 14.

2 You will need to refer to the list of pseudocode string-handling functions in the **Appendix**.

(a) Give the value of the variables `x`, `y` and `z` for the following sequence of statements.

(i)
```
x ← ONECHAR("Barcelona", 5)
```
x ..... [1]

(ii)
```
y ← ONECHAR("Cool", 1) & ONECHAR("Ball", 2) & "t-food"
```
y ..... [1]

(iii)
```
Temp1 ← "13"
Temp2 ← ONECHAR("One-2-One", 5)
z ← TONUM(Temp2 & Temp1)
```
z ..... [1]

A computer program is to simulate the reading and processing of a string of characters from an input device.

The character string consists of:

- a number of digit characters
- one or more <*> characters, each used as a separator
- a final <#> character.

A typical input character sequence, stored as `InputString` is:

```
13*156*9*86*1463*18*#
```

Study this pseudocode.

```
01  DECLARE Numbers ARRAY [1:100] OF INTEGER
02  DECLARE InputString      : STRING
03  DECLARE NextChar         : CHAR
04  DECLARE NextNumberString : STRING
05  DECLARE i                : INTEGER    // Numbers array index
06  DECLARE j                : INTEGER    // InputString index
07
08  OUTPUT "String ... "
09  INPUT InputString
10  j ← 1
11  NextChar ← ONECHAR(InputString, j)
12
13  i ← 1
14  WHILE NextChar <> '#'
15    NextNumberString = ""
16    WHILE NextChar <> '*'
17      NextNumberString ← NextNumberString & NextChar
18      j ← j + 1
19      NextChar ← ONECHAR(InputString, j)
20    ENDWHILE
21
22    // store the next integer to the array
23    Numbers[i] ← TONUM(NextNumberString)
24    i ← i + 1
25    j ← j + 1
26    NextChar ← ONECHAR(InputString, j)
27  ENDWHILE
28
29  CALL DisplayArray()
```

(b) Write the line number for:

(i) A statement which declares a global variable used to store a single character. ..... [1]

(ii) A statement which runs code written as a procedure. ..... [1]

(iii) A statement which indicates the start of a 'pre-condition' loop. ..... [1]

(iv) A statement which increments a variable. ..... [1]

(c) Copy the condition which is used to control the inner loop.

..... [1]

(d) (i) Complete the trace table below for the given pseudocode as far as line 2

The input string is: `23*731*5*#`

i	j	NextChar	NextNumberString	Numbers		
				1	2	3
1	1	'2'				
			""			
			"2"			
	2	'3'	"23"			
	3	'*'		23		

[5]

(ii) Explain what this algorithm does.

.....

.....

.....

..... [2]

QUESTION 15.

2 The following pseudocode represents a simple algorithm.

```
DECLARE NumberFound, Remainder, Number : INTEGER
DECLARE StartNumber, EndNumber, Divisor : INTEGER

INPUT StartNumber
INPUT EndNumber
INPUT Divisor
NumberFound ← 0

FOR Number ← StartNumber TO EndNumber
   Remainder ← MODULUS(Number, Divisor)
   IF Remainder = 0
      THEN
         OUTPUT Number
         NumberFound ← NumberFound + 1
   ENDIF
ENDFOR
OUTPUT "Count: " & NumberFound
```

For the built-in functions list, refer to the **Appendix** on page 14.

(a) Complete the following trace table.

StartNumber	EndNumber	Divisor	NumberFound	Number	Remainder	Output
11	13	2	0			

[3]

(b) Describe the purpose of this algorithm.

.....

.....

.....

.....

.....

.....[3]

(c) Draw a program flowchart to represent this algorithm.

Variable declarations are **not** required in program flowcharts.

[10]

QUESTION 16.

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 17.

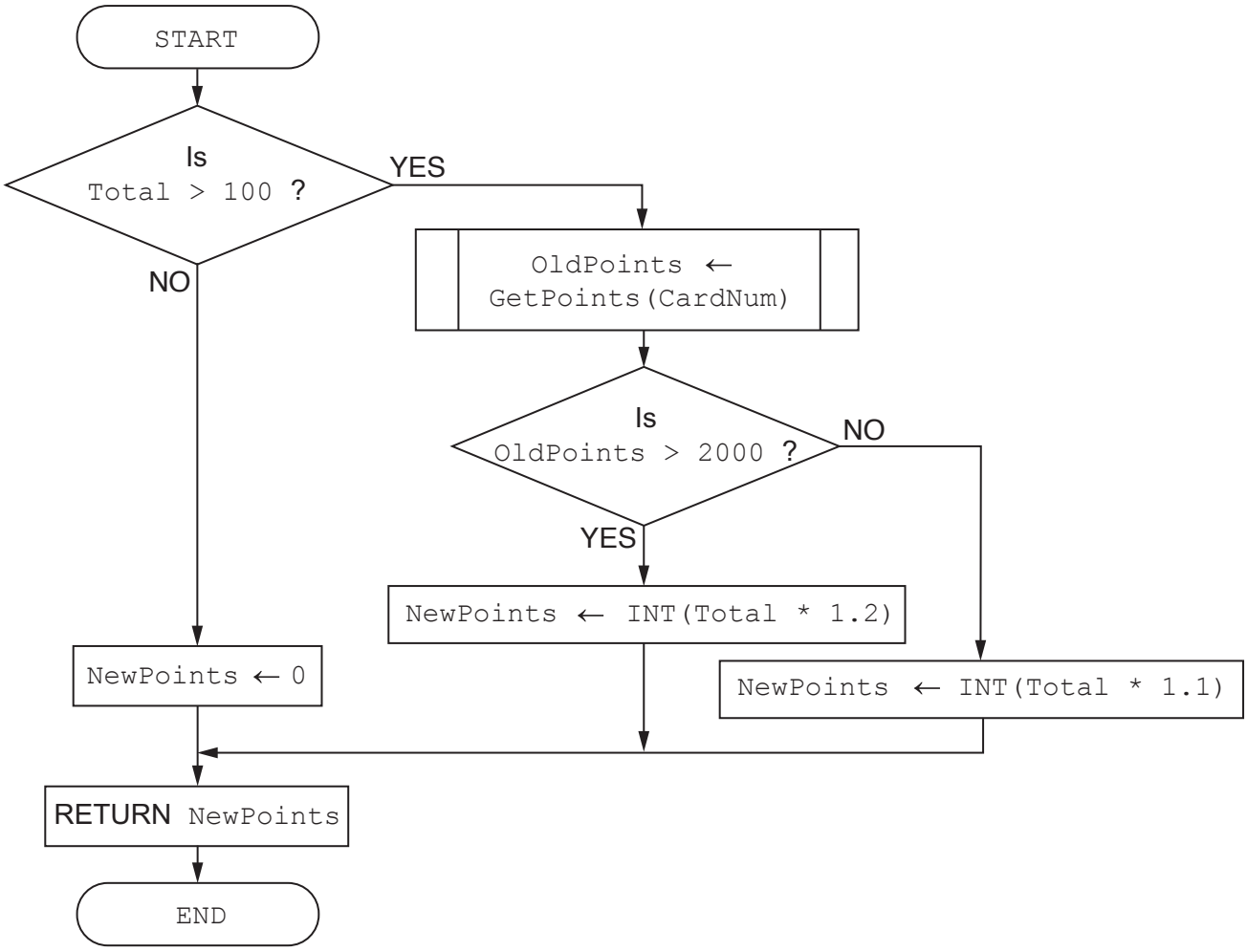
2 Shop customers have a discount card with a unique card number. Customers co
time they buy items. The number of points they collect depends on:

- 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]

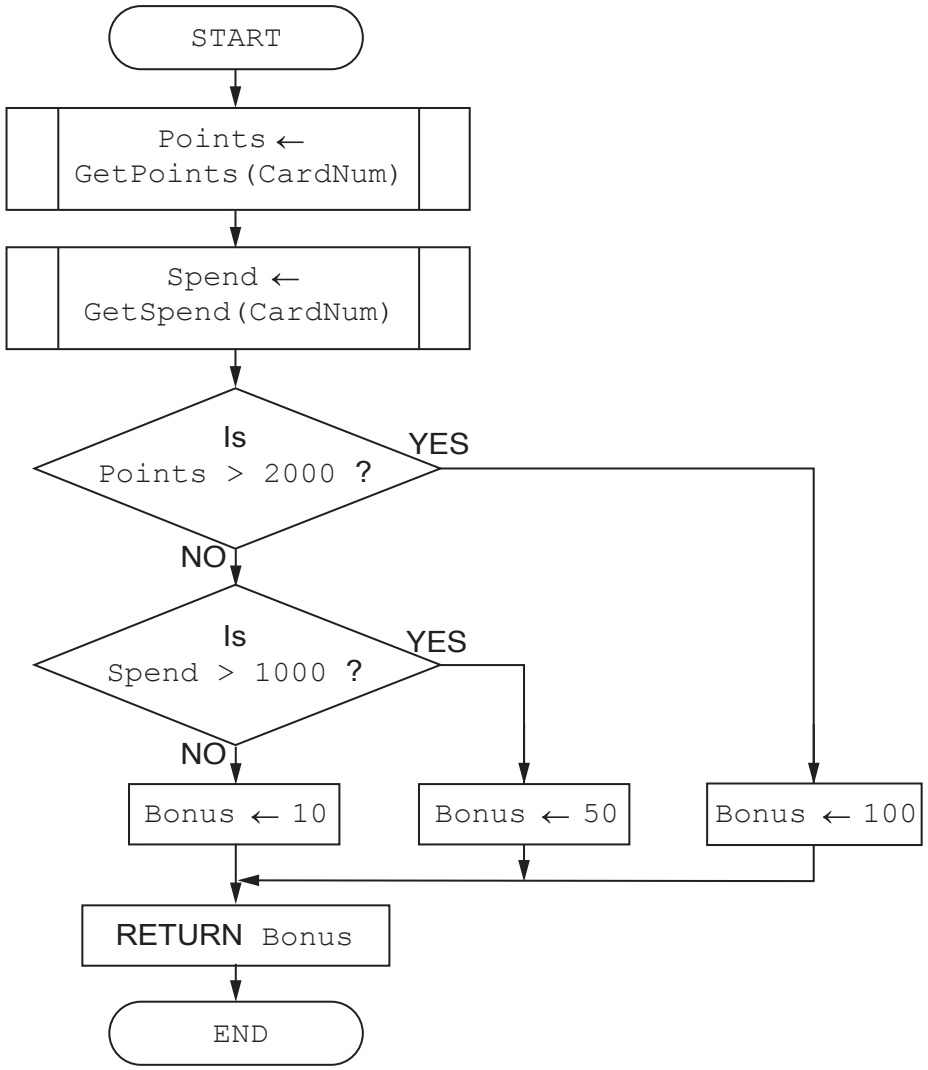
QUESTION 18.

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 19.

4 The following pseudocode is a string handling function.

For the built-in functions list, refer to the **Appendix** on page 16.

```
FUNCTION Clean(InString : STRING) RETURNS STRING

   DECLARE NewString : STRING
   DECLARE Index : INTEGER
   DECLARE AfterSpace : BOOLEAN
   DECLARE NextChar : CHAR
   CONSTANT Space = ' '

   AfterSpace ← FALSE
   NewString ← ""

   FOR Index ← 1 TO LENGTH(InString)
      NextChar ← MID(InString, Index, 1)
      IF AfterSpace = TRUE
         THEN
            IF NextChar <> Space
               THEN
                  NewString ← NewString & NextChar
                  AfterSpace ← FALSE
            ENDIF
         ELSE
            NewString ← NewString & NextChar
            IF NextChar = Space
               THEN
                  AfterSpace ← TRUE
            ENDIF
      ENDIF
   ENDFOR

   RETURN NewString

ENDFUNCTION
```

(a) (i) Complete the trace table by performing a dry run of the function whe
follows:

```
Result ← Clean("X∇∇∇Y∇and∇∇Z")
```

The symbol `'∇'` represents a space character. Use this symbol to represent a spa
character in the trace table.

Index	AfterSpace	NextChar	NewString

[6]

(ii) State the effect of the function `Clean()`.

.....

..... [1]

(iii) The pseudocode is changed so that the variable `AfterSpace` is initialis

Explain what will happen if the function is called as follows:

```
Result ← Clean("∇∇X∇∇∇Y∇and∇∇Z")
```

.....

.....

.....

..... [2]

(b) The following pseudocode declares and initialises an array.

```
DECLARE Code : ARRAY[1:100] OF STRING
DECLARE Index : INTEGER

FOR Index ← 1 TO 100
   Code[Index] ← ""
ENDFOR
```

The design of the program is changed as follows:

- the array needs to be two dimensional, with 500 rows and 4 columns
- the elements of the array need to be initialised to the string `"Empty"`

Re-write the **pseudocode** to implement the new design.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) State the term used for changes that are made to a program in response to a specification change.

..... [1]

Question 5 begins on the next page.

QUESTION 20.

4 The following is pseudocode for a string handling function.

For the built-in functions list, refer to the **Appendix** on page 16.

```
FUNCTION Search(InString : STRING) RETURNS INTEGER

   DECLARE NewString : STRING
   DECLARE Index : INTEGER
   DECLARE NextChar : CHAR
   DECLARE Selected : INTEGER
   DECLARE NewValue : INTEGER

   NewString ← '0'
   Selected ← 0

   FOR Index ← 1 TO LENGTH(InString)

      NextChar ← MID(InString, Index, 1)
      IF NextChar < '0' OR NextChar > '9'
         THEN
            NewValue ← STRING_TO_NUM(NewString)
            IF NewValue > Selected
               THEN
                  Selected ← NewValue
            ENDIF
            NewString ← '0'
         ELSE
            NewString ← NewString & NextChar
      ENDIF

   ENDFOR

   RETURN Selected

ENDFUNCTION
```

(a) (i) The following assignment calls the `Search()` function:

```
Result ← Search("12∇34∇5∇∇39")
```

Complete the following trace table by performing a dry run of this function call.

The symbol `'∇'` represents a space character. Use this symbol to represent a space character in the trace table.

Index	NextChar	Selected	NewValue	NewString

[5]

(ii) State the value returned by the function when it is called as shown in **part (a)(i)**.

.....

[1]

(b) There is an error in the algorithm. When called as shown in **part (a)(i)**, the
return the largest value as expected.

(i) Explain why this error occurred when the program called the function.

.....

.....

.....

..... [2]

(ii) Describe how the algorithm could be amended to correct the error.

.....

.....

.....

QUESTION 21.

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 22.

5 The following pseudocode checks whether a string is a valid password.

```
FUNCTION CheckPassword(InString : STRING) RETURNS BOOLEAN

   DECLARE Index, Upper, Lower, Digit, Other : INTEGER
   DECLARE NextChar : CHAR

   Upper ← 0
   Lower ← 0
   Digit ← 0
   Other ← 0

   FOR Index ← 1 TO LENGTH(InString)

      NextChar ← MID(InString, Index, 1)
      IF NextChar >= 'A' AND NextChar <= 'Z'
         THEN
            Upper ← Upper + 1
         ELSE
         IF NextChar >= 'a' AND NextChar <= 'z'
            THEN
               Lower ← Lower + 1
            ELSE
               IF NextChar >= '0' AND NextChar <= '9'
                  THEN
                     Digit ← Digit + 1
                  ELSE
                     Other ← Other + 1
               ENDIF
         ENDIF
      ENDIF

   ENDFOR

   IF Upper > 1 AND Lower >= 5 AND (Digit - Other) > 0
      THEN
         RETURN TRUE
      ELSE
         RETURN FALSE
   ENDIF

ENDFUNCTION
```

(a) Describe the validation rules that are implemented by this pseudocode. Refer **only** to the contents of the string and **not** to features of the pseudocode.

.....

.....

.....

.....

.....

..... [3]

(b) (i) Complete the trace table by dry running the function when it is called as

```
Result ← CheckPassword("Jim+Smith*99")
```

Index	NextChar	Upper	Lower	Digit	Other

[5]

(ii) State the value returned when the function is called using the expression shown. Justify your answer.

Value .....

Justification .....

.....

.....

[2]

QUESTION 23.

4 The following pseudocode algorithm checks whether a string is a valid email add

```
FUNCTION Check(InString : STRING) RETURNS BOOLEAN

   DECLARE Index : INTEGER
   DECLARE NumDots : INTEGER
   DECLARE NumAts : INTEGER
   DECLARE NextChar : CHAR
   DECLARE NumOthers : INTEGER

   NumDots ← 0
   NumAts ← 0
   NumOthers ← 0

   FOR Index ← 1 TO LENGTH(InString)

      NextChar ← MID(InString, Index, 1)
      CASE OF NextChar
         '.': NumDots ← NumDots + 1
         '@': NumAts ← NumAts + 1
         OTHERWISE NumOthers ← NumOthers + 1
      ENDCASE

   ENDFOR

   IF (NumDots >= 1 AND NumAts = 1 AND NumOthers > 5)
      THEN
         RETURN TRUE
      ELSE
         RETURN FALSE
   ENDIF

ENDFUNCTION
```

(a) Describe the validation rules that are implemented by this pseudocode. Refer **only** to the contents of the string and **not** to features of the pseudocode.

.....

.....

.....

..... [3]

(b) (i) Complete the trace table by dry running the function when it is called as

```
Result ← Check("Jim.99@skail.com")
```

Index	NextChar	NumDots	NumAts	NumOthers

[5]

(ii) State the value returned when function `Check` is called as shown in **part (b)(i)**.

..... [1]

(c) The function `Check()` is to be tested.

State **two** different invalid string values that could be used to test the algorithm. Each should test a different rule.

Justify your choices.

Value .....

Justification .....

.....

.....

Value .....

Justification .....

.....