4 Study the algorithm:

```
DECLARE Chars : ARRAY[1:4] OF CHAR
DECLARE I, J : INTEGER
DECLARE Key : CHAR
I ← 2
Chars[1] ← 'D'
Chars[2] ← 'T'
Chars[3] ← 'H'
Chars[4] ← 'R'
WHILE I <= 4                          //Outer loop
   Key ← Chars[I]
   J ← I - 1
   WHILE J >= 0 AND Chars[J] > Key    //Inner loop
      Chars[J + 1] ← Chars[J]
      J ← J - 1
   ENDWHILE
   Chars[J + 1] ← Key
   I ← I + 1
ENDWHILE
```

(a) The outer loop structure used in the algorithm is **not** the most appropriate one to use.

State the type of loop structure that would be the most appropriate to use and justify why it is the most appropriate.

Loop structure .....

Justification .....

.....

[2]

(b) Complete the trace table by dry running the algorithm.

The first row has been completed.

I	Key	J	Chars[J]	Chars			
				[1]	[2]	[3]	[4]
2				'D'	'T'	'H'	'R'
				②			

[6]

2 A program uses three global integer variables `HH`, `MM` and `SS` to represent the current time in hours, minutes and seconds using the 24-hour clock notation.

Midnight would be represented as 00:00:00 (`HH:MM:SS`). If the variables `HH`, `MM` and `SS` contained the values 16, 30 and 10 respectively, then the time would be 16:30:10 or just after 4.30 in the afternoon.

A procedure `Tick()` will be called every second.
The procedure `Tick()` will:

- update the value in `SS` each time it is called
- update the values in `HH` and `MM` as appropriate
- call a procedure `CheckAlarm()` at the start of each minute
- call a procedure `NewDay()` whenever the time reaches midnight.

Complete the pseudocode for procedure `Tick()`.

```
PROCEDURE Tick()
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
ENDPROCEDURE
```


[6]

5 A program contains a global 1D array `Data` with 100 elements of type `INTEGER`.

The program contains a function `Process()` expressed in pseudocode as follows:

```
FUNCTION Process(Number : INTEGER, Label : STRING) RETURNS STRING
   DECLARE Index, Count : INTEGER
   DECLARE ReturnValue : STRING

   Count ← INT(100 / Number)
   Index ← Data[Number]

   CASE OF (Index MOD 2)
      0 : ReturnValue ← TO_UPPER(RIGHT(Label, Count))
      1 : ReturnValue ← "*****"
   ENDCASE

   RETURN RetVal
ENDFUNCTION
```

(a) Run-time errors can be generated in different ways. For example, a run-time error will be generated if a function is called with invalid parameters.

The pseudocode contains three statements that could generate a run-time error.

Write the **three** statements **and** explain how each could generate a run-time error.

Statement 1 .....

Explanation .....

.....

Statement 2 .....

Explanation .....

.....

Statement 3 .....

Explanation .....

.....

[3]

(b) One type of run-time error can cause a program to stop responding ('freezing').

Identify a particular type of programming construct that can generate this type of error **and** explain why it occurs.

Construct .....

Explanation .....

.....

.....
[2]

(c) The function `Process()` contains a selection construct using a `CASE` structure.

Write pseudocode using a single selection construct with the same functionality **without** using a `CASE` structure.

.....

.....

.....

.....

..... [2]

2 An algorithm will:

1. prompt and input a sequence of 100 integer values, one at a time
2. sum the positive integers
3. output the result of the sum.

(a) Write pseudocode for the algorithm.

Assume the value zero is neither positive nor negative.

You must declare all variables used in the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) The algorithm requires the use of basic constructs. One of these is sequence.

Identify **one other** basic construct required by the algorithm **and** describe how it is used.

Construct .....

Use .....

.....
[2]

4 An examination paper has a maximum of 75 marks. One of five pass grades (A to E) is assigned, depending on the mark obtained. The lowest mark for a given grade is known as the grade boundary.

A program is being written to process examination marks.

The five grade boundaries are stored in a global 1D array `GB` of type `INTEGER`, for example:

Index	Value	Comment
1	65	The minimum mark for an A grade.
2	57	The minimum mark for a B grade.
3	43	The minimum mark for a C grade.
4	35	The minimum mark for a D grade.
5	27	The minimum mark for an E grade.

Any paper that achieves a mark within 2 marks of a grade boundary must be checked. Using the given table, a paper with 45 marks would need to be checked.

(a) The pseudocode algorithm to determine whether a paper should be checked is as shown. The mark for the paper is stored in variable `Mark`. Global variables `Mark`, `Index`, `Upper` and `Lower` are declared as integers.

Complete the pseudocode.

```
FOR Index ← 1 TO .....
   Lower ← GB[Index] - 2
   Upper ← .....
   IF Mark ..... AND Mark ..... THEN
      OUTPUT "Check this paper"
   ENDIF
NEXT Index
```

[4]

(b) An alternative algorithm to determine if a paper needs to be checked uses a global 1D array `Check`, containing 76 elements of type `BOOLEAN`. The indices of the array are from 0 to 75 (inclusive), corresponding to the range of possible marks.

An element value in `Check` is `TRUE` if the index is within 2 marks of a grade boundary. For example, in the case where the C grade boundary is 43 the corresponding part of the `Check` array would be as follows:

Index	Value	
40	FALSE	
41	TRUE	
42	TRUE	
43	TRUE	← The grade boundary for a C grade
44	TRUE	
45	TRUE	
46	FALSE	

(i) The mark for a given paper is stored in variable `Mark`.

Describe how an algorithm would use the `Check` array to determine whether this paper should be checked.

.....

.....

.....

..... [2]

(ii) A procedure `GBInitialise()` will initialise the `Check` array using values from the `GB` array.

Note it can be assumed that the maximum grade boundary value for A is 70 and the minimum value for E is 15.

Write pseudocode for the procedure.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

5 A global 1D array of strings contains three elements which are assigned values as shown:

```
Data[1] ← "aaaaaa"
Data[2] ← "bbbbbb"
Data[3] ← "cccccc"
```

Procedure `Process()` manipulates the values in the array.

The procedure is written in pseudocode as follows:

```
PROCEDURE Process(Format : STRING)
    DECLARE Count, Index, L : INTEGER
    DECLARE Result : STRING
    DECLARE C : CHAR

    Result ← "****"

    FOR Count ← 1 TO LENGTH(Format) STEP 2
        C ← MID(Format, Count, 1)
        L ← STR_TO_NUM(MID(Format, Count + 1, 1))

        Index ← (Count + 1) DIV 2

        CASE OF C
            'X' : Result ← TO_UPPER(Data[Index])
            'Y' : Result ← TO_LOWER(Data[Index])
            'Z' : Result ← "**" & Data[Index]
        ENDCASE

        Data[Index] ← LEFT(Result, L)
    NEXT Count

ENDPROCEDURE
```


(a) Complete the trace table by dry running the procedure when it is called as follows:

```
CALL Process("X3Y2W4")
```

Count	C	L	Index	Result	Data[1]	Data[2]	Data[3]

[6]

(b) The procedure is to be modified. If variable `C` is assigned a value other than `'X'`, `'Y'` or `'Z'`, then procedure `Error()` is called and passed the value of variable `C` as a parameter.

This modification can be implemented by adding a **single line** of pseudocode.

(i) Write the single line of pseudocode.

..... [1]

(ii) State where this new line should be placed.

..... [1]

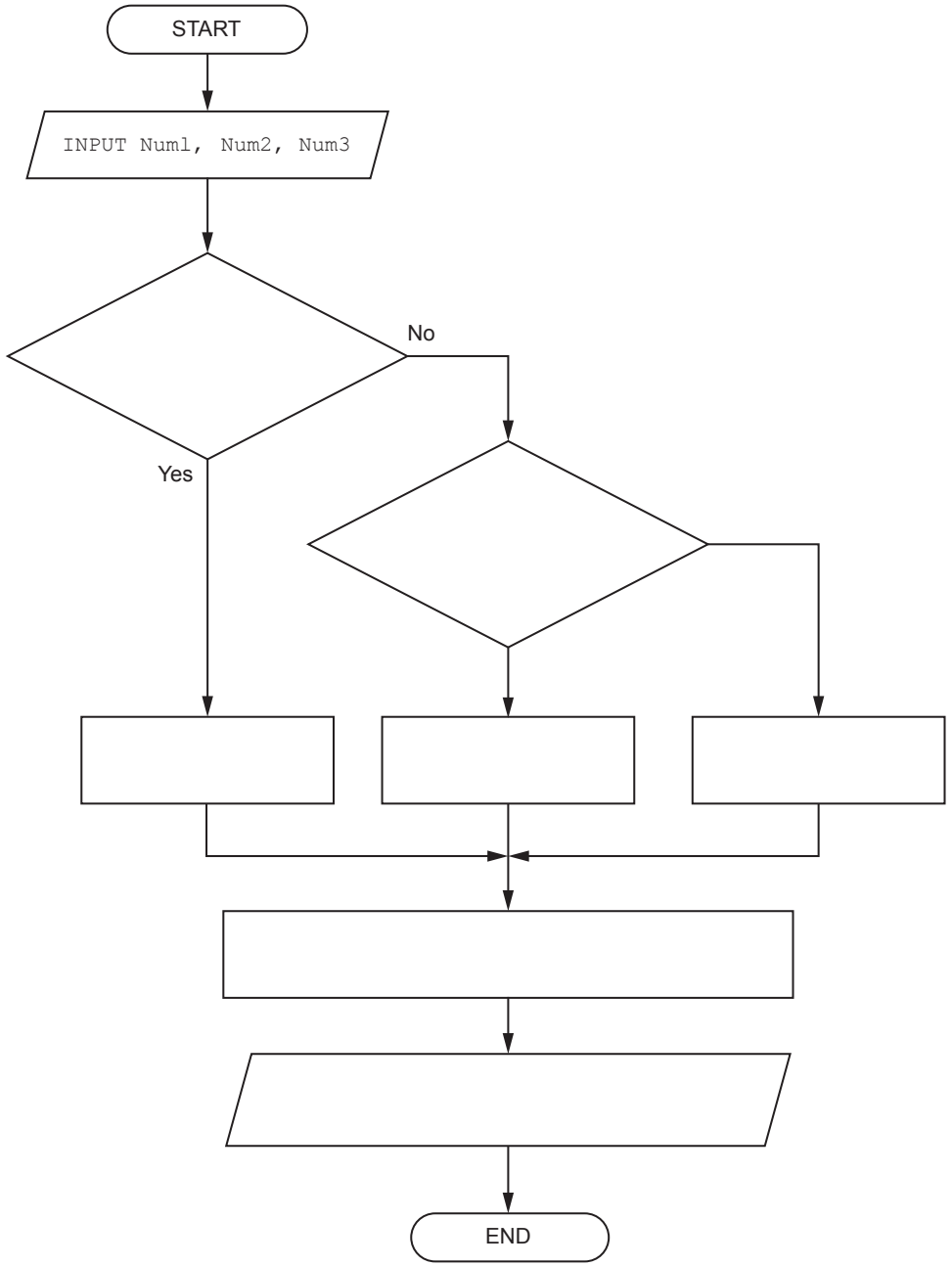
2 A program is being developed.

(a) An algorithm for part of the program will:

- input three numeric values and assign them to identifiers `Num1`, `Num2` and `Num3`
- assign the largest value to variable `Ans`
- output a message giving the largest value and the average of the three numeric values.

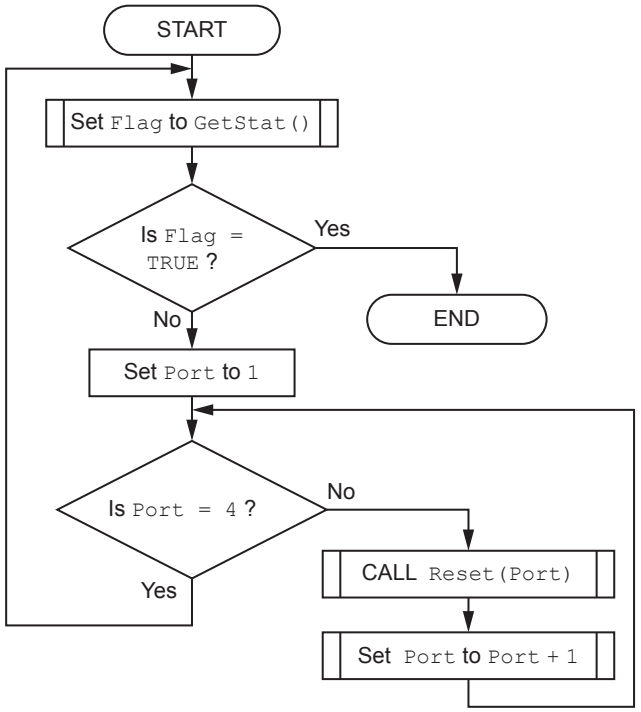
Assume the values are all different and are input in no particular order.

Complete the program flowchart on page 5 to represent the algorithm.



[5]

(b) A different part of the program contains an algorithm represented by the following program flowchart:



Write pseudocode for the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

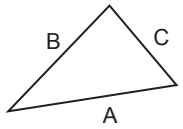
.....

.....

.....

[5]

4 A triangle has sides of length A, B and C.



In this example, A is the length of the longest side.

This triangle is said to be right-angled if the following equation is true:

$A \times A = (B \times B) + (C \times C)$

A procedure will be written to check whether three lengths represent a right-angled triangle. The lengths will be input in any sequence.

The procedure `IsRA()` will:

- prompt and input three integer values representing the three lengths
- test whether the three lengths correspond to the sides of a right-angled triangle
- output a suitable message.

The length of the longest side may **not** be the first value input.

Write pseudocode for the procedure `IsRA()`.

....

.... [5]

5 A program is being designed in pseudocode.

The program contains a global 1D array `Data` of type string containing 200 elements.

The first element has the index value 1.

A procedure `Process()` is written to initialise the values in the array:

```
PROCEDURE Process(Label : STRING)
   DECLARE Index : INTEGER
   Index ← 0
   INPUT Data[Index]
   WHILE Index < 200
      Index ← Index + 1
      CASE OF (Index MOD 2)
         0 : Data[Index] ← TO_UPPER(Label)
         1 : Data[Index] ← TO_LOWER(Label)
         OTHERWISE : OUTPUT "Alarm 1201"
      ENDCASE
   NEXT Index
   OUTPUT "Completed " & Index & " times"
ENDPROCEDURE
```

(a) (i) The pseudocode contains **two** syntax errors and **one** other error.

Identify the errors.

Syntax error 1 ....

....

Syntax error 2 ....

....

Other error ....

.... [3]

(ii) The procedure contains a statement that is **not** needed.

Identify the pseudocode statement **and** explain why it is **not** needed.

Statement ....

Explanation ....

.... [2]

(b) After correcting all syntax errors, the pseudocode is translated into program code which compiles without generating any errors.

When the program is executed it unexpectedly stops responding.

Identify the type of error that has occurred.

..... [1]