6 A program monitors the speed of vehicles as they move around a large building site.

Each vehicle contains a sensor which reads an integer value that represents the speed of the vehicle. The value is expected to be in the range 0 to 60 inclusive.

The sensors cannot read values less than 0.

A program module has been written to validate the values read by the sensors.

(a) A test plan is needed to fully test the module.

Complete the table. The first line has been completed for you.

Assume the sensors generate only integer values.

Type of test data	Test data value	Expected outcome
normal	36	data item is accepted

[4]

(b) Each sensor used in the system has a unique sensor ID number in the range 1 to 50.

The program that is used to monitor the speed of each vehicle reads each sensor value every second and stores this value along with the sensor's unique ID into a global 2D array `Reading`

The global array `Reading` has been declared as follows:

```
DECLARE Reading : ARRAY[1:2000, 1:2] OF INTEGER
```

The array contains 4000 elements organised as 2000 rows and 2 columns.

Column 1 contains the sensor value, and column 2 contains the sensor ID.

When 2000 sensor readings have been taken, the array is full and the system stops taking any more sensor readings until the array has been processed.

A procedure `Sort()` is needed to sort the array into ascending order of sensor value using an efficient bubble sort algorithm.

Write **efficient** pseudocode for the procedure `Sort()`

[8]

2 A program to calculate the pay of employees working for a company is being designed.

(a) Stepwise refinement has been used in the design of the program.

Describe stepwise refinement.

.....

.....

.....

..... [2]

(b) One of the program modules used to calculate employees' weekly bonus pay has been completed. The amount of bonus pay they receive is based on the number of hours worked and the value of sales made as shown in the table.

Bonus pay and value of sales are both in dollars.

Hours worked	Value of sales	Bonus pay
between 1 and 40 inclusive	2000 or less	0
between 1 and 40 inclusive	above 2000	50
above 40	2000 or less	10
above 40	above 2000	100

(i) The module is tested using white-box testing.

State **two** tests, using valid data, that can be used to test different paths through the program.

Test one:

Hours worked .....

Value of sales .....

Bonus pay .....

Test two:

Hours worked .....

Value of sales .....

Bonus pay .....

[2]

(ii) The program to calculate pay uses a number of modules which are each called from different places in the program.

The program is to be tested using stub testing before all the program modules have been completed.

Describe stub testing.

.....

.....

.....

..... [2]

(iii) The program has been completed and compiles successfully.

The program is tested using black-box testing.

Identify and describe **one** type of error that black-box testing could detect.

Type of error .....

Description .....

.....

.....

[2]

1 A program has been developed and released for general use. After a few months of use an error is detected where under certain circumstances it outputs an unexpected value.

(a) The error in the program needs to be corrected.

Identify the stage of the program development life cycle that this correction is made in.

..... [1]

(b) The program contains a function `Lookup()`. After investigation, it is found that this is the function that sometimes returns an incorrect value.

An Integrated Development Environment (IDE) is used to help locate the error.

The IDE features of watch window, single stepping and breakpoint will be used.

Explain these features including the order that they will be used in to locate the error in `Lookup()`.

.....

.....

.....

.....

.....

..... [3]

(c) To solve the error a programmer decides to create a new module.

The design of the new module has been completed and the module is being coded.

Identify **two** features of an IDE that will help during the coding of this new module.

1 .....

2 .....

[2]

(d) The new module referred to in part **(c)** introduces **three** new variables.

Complete the following table by giving the appropriate data type for each.

Variable name	Used to store	Data type
`Name`	A customer name.	
`Index`	An array index.	
`Result`	The result of the division of any two non-zero numbers.	

[3]

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]

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]

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]