31 mins 9 questions

Exam Questions

Program Testing & Maintenance

Faults & errors / Testing methods / Maintenance

1 Refer to the insert for the list of pseudocode functions and operators.

A program will calculate the tax payable based on the cost of an item.

Calculations will occur at many places in the program and these involve the use of one of three tax rates.

Tax rate values represent a percentage. For example, a tax rate value of 5.23 represents 5.23%. In this case, the tax payable on an item costing $100 would be $5.23.

Tax rate values are used at several places within the program. One example is given in pseudocode as follows:

```
HighRate ← FALSE

CASE OF ItemCost

<= 50 : TaxRate ← 3.75 // tax rate of 3.75%

<= 200 : TaxRate ← 5.23 // tax rate of 5.23%

> 200 : TaxRate ← 6.25 // tax rate of 6.25%

HighRate ← TRUE

ENDCASE

TaxPayable ← ItemCost * TaxRate // tax payable
```

The pseudocode contains a logical error.

Identify the error **and** suggest a correction.

Answer

Mark Scheme and Guidance

One mark per point:

Error: The **calculation** of TaxPayable is incorrect

Correction: TaxPayable (ItemCost * TaxRate) / 100

(2 marks)

2 A program includes the following assignment statement:

Result ← STR_TO_NUM(x) / STR_TO_NUM(y)

When the program evaluates the expression in the statement, it performs a calculation.

Variable Result is of type real and variables x and y are of type string.

Two checks are required before the calculation is performed:

Identify the type of error that could occur if these checks are **not** carried out **and** state a cause of this error.

Answer

Mark Scheme and Guidance

One mark per point:

Type: Run-time error

Cause: A divide by zero operation is attempted // Attempt to convert a nonnumeric string to a number

(2 marks)

3 A software developer follows a program development life cycle. The life cycle divides the development process into various stages.

The program contains a validation function.

(i) The function will:

- take an integer value as a parameter .

- return TRUE if the value is within the range 24 to 37, inclusive
- otherwise return FALSE.

Complete the table to define a test plan to thoroughly test the operation of the function.

Type of test data	Test data value	Expected result
Normal	30	TRUE

(4)

(ii) The function is to be tested on its own. When it is shown to work correctly the function will be combined with other modules and testing will continue.

Identify the type of testing that this represents.

(1)

Answer

Mark Scheme and Guidance

(i) One mark per row

Type of test data	Test data value	Expected result
Abnormal	12 (< 23)	FALSE
Abnormal / Boundary / Extreme	23	FALSE
Boundary / Extreme	24	TRUE
Boundary	25	TRUE
Boundary	36	TRUE
Boundary /Extreme	37	TRUE
Abnormal / Boundary / Extreme	38	FALSE
Abnormal	99 (> 38)	FALSE

Max 4 marks

(ii) Integration (testing)

(5 marks)

4 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
```

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

Answer

Mark Scheme and Guidance

MP1 Count ← INT(100 / Number)
Number could be zero (giving a divide by zero)

MP2 Index ← Data[Number]
Potential error: Value Number could be outside the range of array indices

MP3 ReturnValue ← TO_UPPER(RIGHT(Label, Count))
Potential Error: Number to extract may be too big / negative / out of range for use in the RIGHT function // Label has insufficient characters

MP4 RETURN RetVal
Potential Error: There is no value to be returned // there is no variable named RetVal

Mark as follows:
1 mark for each statement **and** description
Max 3 marks

(3 marks)

5 A software developer follows a program development life cycle. The life cycle divides the development process into various stages.

A software developer has written modules Test_A() and Test_B(). These have been written but contain errors. These modules are called from several places in the main program and testing of the main program (integration testing) has to stop.

Identify a method that can be used to continue testing the main program **before** the errors in these modules have been corrected **and** describe how this would work.

Answer

Mark Scheme and Guidance

One mark for method – two marks for the description

MP1 Stub testing

MP2 The modules Test_A() and Test_B()are replaced by dummy modules

MP3 … which return a known result // An output statement is displayed when called (to check it works) // gives expected output

(3 marks)

6 (a) 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
```

The pseudocode contains **two** syntax errors and **one** other error.

Identify the errors.

Answer

Mark Scheme and Guidance

One mark per error:

Syntax:

1. NEXT Index (should be ENDWHILE)

2. '&' used to concatenate an integer (in OUTPUT statement)

Other:

3. Accesses element outside range // Accesses element 0

(3 marks)

(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.

Answer

Mark Scheme and Guidance

Run-time

(1 mark)

7 (a) An algorithm is designed to find the smallest numeric value from an input sequence and count how many numeric values have been input.
An example of an input sequence is:

23, AB56, 17, 23ZW, 4, 10, END

Numeric input values are all integers and non-numeric input is ignored, except for the string "END" which is used to terminate the sequence.

The algorithm is expressed in pseudocode as shown:

```
DECLARE NextInput : STRING
DECLARE Min, Count, Num : INTEGER

Min ← 999
Count ← 0

REPEAT
INPUT NextInput
IF IS_NUM(NextInput) = TRUE THEN
Num ← STR_TO_NUM(NextInput)
IF Num > Min THEN
Min ← Num
ENDIF Count ← Count & 1
ENDIF
UNTIL NextInput ← "END"

OUTPUT "The minimum value is ", Min, " and the count was ", Count
```

The pseudocode contains three errors due to the incorrect use of **operators.**

Identify each error **and** state the correction required.

Answer

Mark Scheme and Guidance

- **MP1** Num > Min should be Num < Min
- **MP2** Count & 1 should be Count + 1

- **MP3** NextInput "END" should be NextInput = "END"

(3 marks)

(b) The operator errors are corrected and the algorithm is tested as follows:

The input sequence:

18, 4, ONE, 27, 189, ERIC, 3, 65, END

produces the output:

The minimum value is 3 and the count was 6

The algorithm is tested with a different test data sequence. The sequence contains a mix of integer and non-numeric values. It is terminated correctly but the algorithm produces unexpected results.

(i) Explain the problem with the algorithm

[2]

(ii) Give a sequence of four test data values that could be input to demonstrate the problem.

Value 1 ..... Value 2 ..... Value 3 ..... Value 4 .....

[2]

Answer

Mark Scheme and Guidance

(i)

- **MP1** If all the numeric input values are greater than 999 // If there are no numeric values in the sequence
- **MP2** then the minimum will be given as **999** (and not one of the input values)

(ii)

Many possible correct answers, for example:

- **MP1** Mixture non-numeric and numeric with 3 or 4 values - with all numerics greater than 999
  Examples: 1325, DOG, 7868, 7615
  // SNAKE, 3478, SPIDER
- **MP2** Final value: END

(4 marks)

8 A procedure Count() will:

1. input a value (all values will be positive integers)
2. count the number of odd values and count the number of even values
3. repeat from step 1 until the value input is 99
4. output the two count values, with a suitable message.

The value 99 must not be counted

The procedure Count() is to be tested.

Typical test data would consist of odd and even values, for example:

23, 5, 64, 100, 2002, 1, 8, 900, 99

The purpose of this test would be to test a typical mix of even and odd values and check the totals.

Give **three** test data sequences that may be used to test **different** aspects of the procedure.

Do **not** include invalid da.

Sequence 1:

Test data .....

Purpose of test. .....

.....

Sequence 2:

Test data .....

Purpose of test. .....

.....

Sequence 3:

Test data .....

Purpose of test. .....

.....

Answer

Mark Scheme and Guidance

One mark per set, including stated purpose. Max 3 marks

Example answers:

1 data set with (only) odd values, terminated with 99
2 data set with (only) even values, terminated with 99
3 data sets with same number of odd and even values, terminated with 99
4 data sets with all even / all odd with just one odd/even value, terminated with 99
5 data sets with no values just final 99
6 data sets without (terminating) 99 // missing or incorrectly placed 99

(3 marks)

9 A program is being developed in pseudocode before being converted into a programming language.

The hardware that runs the program is changed and the program needs to be modified so that it works with the new hardware.

Identify the type of maintenance that this represents and give one other reason why this type of maintenance may be needed.

Type ..... Reason .....
.....

Answer

Mark Scheme and Guidance

- **MP1** Type: Adaptive
- **MP2** Reason: The (user) requirement(s) changes // to accommodate legislative changes

(2 marks)