

AS · Cambridge (CIE) · Computer Science

🕒 33 mins 🗋️ 8 questions

Exam Questions

Programming Basics

Pseudocode basics

- 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
```

During the design of the program, tax rate values have been used wherever they are needed as shown in the pseudocode example above. Tax rates do not change while the program runs.

- (i) Identify a more appropriate way of representing the tax rate values in the final program.

[1]

- (ii) Describe the benefits of your answer to part **(b)(i)** with reference to this program.

[3]

Answer



Mark Scheme and Guidance

- (i)

Use constants (to represent the tax rate values)

(ii)

One mark per bullet point (or equivalent to max 3):

1. **Tax rates** are entered once only
2. Avoids / Minimise (input) error(s) / changing the Tax rates accidentally // avoids different values for tax rates at different points in the program
3. When required, the constant value (representing a tax rate) is changed (once) // Easier to maintain / update the program (when the tax rates change)
4. Makes the program / code easier to understand

(4 marks)

2 Evaluate each expression in the table by using the data values shown in part (b).

Write 'ERROR' if the expression contains an error.

Expression	Evaluates to
Available AND NOT(Index > 100)	
Index MOD 30	
NUM_TO_STR(Index + "33")	

Answer



Mark Scheme and Guidance

Expression	Evaluates to
Available AND NOT(Index > 100)	TRUE
Index MOD 30	10
NUM_TO_STR(Index + "33")	ERROR

One mark per row

(3 marks)

3 (a) Refer to the [insert](#) for the list of pseudocode functions and operators.

An algorithm is developed in pseudocode before being coded in a programming language.

The following table shows four valid pseudocode assignment statements. Complete the table by giving an appropriate data type to declare each of the variables **A**, **B**, **C** and **D**.

Assignment statement	Data type
A ← LEFT(MyName, 1)	
B ← Total * 2	
C ← INT(ItemCost) / 3	
D ← "Odd OR Even"	

Answer



Mark Scheme and Guidance

Assignment statement	Data type
A ← LEFT(MyName, 1)	CHAR / STRING
B ← Total * 2	INTEGER / REAL
C ← INT(ItemCost) / 3	REAL
D ← "Odd OR Even"	STRING

One mark per row

(4 marks)

(b) Other variables in the program have example values as shown:

Variable	Value
Sorted	False
Tries	9
ID	"ZGAC001"

Complete the table by evaluating each expression, using the example values.

Expression	Evaluates to
Tries < 10 AND NOT Sorted	
Tries MOD 4	
TO_LOWER(MID(ID, 3, 1))	
LENGTH(ID & "xx") >= Tries	

Answer



Mark Scheme and Guidance

Expression	Evaluates to
Tries < 10 AND NOT Sorted	TRUE
Tries MOD 4	1
TO_LOWER(MID(ID, 3, 1))	'a' // "a"
LENGTH(ID & "xx") >= Tries	TRUE

One mark per row

(4 marks)

4 An algorithm has three steps. It will:

1. repeatedly input a pair of numeric values A and B
2. count the number of pairs that are input until A has been greater than B 10 times
3. output the number of pairs that were input.

Step 1 of the algorithm is changed.

A variable **ThisSequence** is used to enter a sequence of 10 pairs of numeric values, using a single input statement.

Following the input of **ThisSequence** the revised algorithm will extract the pairs of numbers.

Describe the variable **ThisSequence** and how the numbers are extracted.

Answer



Mark Scheme and Guidance

One mark per point:

1. A (variable of type) string will be input // by example e.g. "67,72"
2. A special / identified character would need to be used to separate each numeric value // all numbers are fixed length

(2 marks)

5 Program variables have data types as follows:

Variable	Data type
MyChar	CHAR
MyString	STRING
MyInt	INTEGER

Complete the table by filling in each gap with a function (from the insert) so that each expression is valid.

Expression
MyInt ← (3.1415926)
MyChar ← ("Elwood", 3, 1)
MyString ← (..... (27.509))
MyInt ← (..... ("ABC123", 3))

Answer



Mark Scheme and Guidance

Expression
MyInt ← INT (3.1415926)
MyChar ← MID ("Elwood", 3, 1)
Any of: <ul style="list-style-type: none"> • MyString ← NUM_TO_STR (INT (27.509)) • MyString ← CHR (INT (27.509)) • MyString ← TO_UPPER(NUM_TO_STR(27.509)) • MyString ← TO_LOWER(NUM_TO_STR(27.509))
Any of: <ul style="list-style-type: none"> • MyInt ← STR_TO_NUM (RIGHT ("ABC123", 3)) • MyInt ← LENGTH (RIGHT ("ABC123", 3)) • MyInt ← LENGTH (LEFT ("ABC123", 3))

One mark per row

(4 marks)

6 Refer to the [insert](#) for the list of pseudocode functions and operators.

Four of the expressions used in the program are represented by pseudocode in the table.

Complete each pseudocode expression with a function or operator so that it evaluates to the value shown.

Any functions and operators used must be defined in the **insert**.

Pseudocode expression	Evaluates to
..... ("Random", 2, 3)	"and"
5 + (10/11/2023)	15
..... ("45000")	TRUE
(20 3) + 1	3

Answer



Mark Scheme and Guidance

Pseudocode expression	Evaluates to
MID ("Random", 2, 3)	"and"
5 + DAY (10/11/2023)	15
IS_NUM ("45000")	TRUE
(20 MOD 3) + 1	3

One mark per row

(4 marks)

7 Other variables in the program have example values as shown:

Variable	Example value
Total	124.00
DepRate	2.00
Description	"AB12345:Cleaning Brush (small)"

Complete the table by evaluating each expression using the example values.

Expression	Evaluates to
$(Total * DepRate) + 1.5$	
<code>RIGHT(Description, 7)</code>	
$(LENGTH(Description) - 8) > 16$	
<code>NUM_TO_STR(INT(DepRate * 10)) & '%'</code>	

Answer



Mark Scheme and Guidance

One mark per row

Expression	Evaluates to
$(Total * DepRate) + 1.5$	249.50
RIGHT(Description, 7)	"(small)"
$(LENGTH(Description) - 8) > 16$	TRUE
NUM_TO_STR(INT(DepRate * 10)) & '%'	"20%"

(4 marks)

8 Other variables in the program have example values as shown:

Variable	Value
Active	TRUE
Fraction	0.2
Code	"Ab12345"

Complete the table by evaluating each expression using the example values.

Expression	Evaluates to
$Fraction \geq 0.2 \text{ AND NOT Active}$	
$INT((Fraction * 100) + 13.3)$	
$STR_TO_NUM(MID(Code, 4, 2)) + 5$	
$LENGTH("TRUE" \& Code)$	

Answer



Mark Scheme and Guidance

Expression	Evaluates to
Fraction >= 0.2 AND NOT Active	FALSE
INT((Fraction * 100) + 13.3)	33
STR_TO_NUM(MID(Code, 4, 2)) + 5	28
LENGTH("TRUE" & Code)	11

(4 marks)