AS · Cambridge (CIE) · Computer Science 57 mins 13 questions

Exam Questions

Data Types & Records

Data types / Record structures

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
```

Give the **appropriate** data type for the variables in the following table, as used in the pseudocode:

Variable name	Data type
HighRate	
TaxPayable	

Answer

Mark Scheme and Guidance

One mark per row:

Variable name	Data type
HighRate	Boolean
TaxPayable	Real

(2 marks)

2 The module to perform the checks and calculation will be implemented as a function. The function will need to return both a real and a Boolean value. To achieve this a record type is defined in pseudocode as follows:

```
TYPE Result
DECLARE Done : BOOLEAN
DECLARE Value : REAL
ENDTYPE
```

The function Evaluate() will:

- take two parameters of type string representing the two numeric values
- return a variable of type Result with the Done field set to FALSE if either of the following applies:
    - at least one of the strings does **not** represent a valid numeric value
    - the numeric value of the string representing value y is zero
- otherwise return a variable of type Result with the Done field set to TRUE and the Value field assigned the result of the formula (based on the numeric value of the two parameters).

Write pseudocode for the function Evaluate().

Answer

Mark Scheme and Guidance

Example solution:

```
FUNCTION Evaluate(NumStr1, NumStr2 : STRING) RETURNS Result
DECLARE RetVal : Result
DECLARE Num1, Num2 : REAL

RetVal.Done ← FALSE

IF IS_NUM(NumStr1) = TRUE AND IS_NUM(NumStr2) = TRUE THEN
Num1 ← STR_TO_NUM(NumStr1)
Num2 ← STR_TO_NUM(NumStr2)
IF Num2 <> 0 THEN
RetVal.Value ← (Num1 / Num2)
RetVal.Done ← TRUE
ENDIF
ENDIF

RETURN RetVal

ENDFUNCTION
```

Note that order of parameters is not specified so divisor could be either parameter.

Mark as follows:

1. Function heading, parameters, ending
2. … Correct return type of Result
3. Local declaration of type Result
4. Attempt at using IS_NUM() to check **both** parameters/strings
5. Attempt at using of STR_TO_NUM() to convert **both** parameters/strings
6. Check if the divisor, is non-zero …
7. … **and** if so correct calculation **and** assignment to Value field
8. Return variable of type Result having assigned both fields

Max 6 marks

(6 marks)

3 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.

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.	

Answer

Mark Scheme and Guidance

One mark per row:

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

(3 marks)

4 A program is being developed to implement a game for up to six players.

During the game, each player assembles a team of characters. At the start of the game there are 45 characters available.

Each character has four attributes, as follows:

Attribute	Examples	Comment
Player	0, 1, 3	The player the character is assigned to
Role	Builder, Teacher, Doctor	The job that the character will perform in the game.
Name	Bill, Lee, Farah, Mo	The name of the character. Several characters may perform the same role, but they will each have a unique name.
Level	14, 23, 76	The skill level of the character. An integer in the range 0 to 100 inclusive.

The programmer has defined a record type to define each character.

The record type definition is shown in pseudocode as follows:

```
TYPE CharacterType
DECLARE Player : INTEGER
DECLARE Role : STRING
DECLARE Name : STRING
DECLARE Level : INTEGER
ENDTYPE
```

The Player field indicates the player to which the character is assigned (1 to 6). The field value is 0 if the character is **not** assigned to any player.

The programmer has defined a global array to store the character data as follows:

```
DECLARE Character : ARRAY[1:45] OF CharacterType
```

At the start of the game all record fields are initialised, and all Player field values are set to 0

The programmer has defined a program module as follows:

Module	Description

Assign()	• called with two parameters:
	  ◦ an integer representing a player
	  ◦ a string representing a character role
	• search the Character array for an unassigned character with the required role
	• If found, assign the character to the given player and output a confirmation message, for example:
	"Bill the Builder has been assigned to player 3"
	• If no unassigned character with the required role is found, output a suitable message.

Write pseudocode for module Assign().

Answer

Mark Scheme and Guidance

Example solution:

```
PROCEDURE Assign(ThisRole : STRING, ThisPlayer : INTEGER)
DECLARE Index : INTEGER
DECLARE Done : BOOLEAN

Done ← FALSE
Index ← 1

WHILE Index < 46 AND Done = FALSE
IF Character[Index].Player = 0 AND __
Character[Index].Role = ThisRole THEN
```

```
Character[Index].Player ← ThisPlayer
Done ← TRUE
ELSE
Index ← Index + 1
ENDIF
ENDWHILE

IF Done = TRUE THEN
OUTPUT Character[Index].Name, " the ", __
Character[Index].Role, __
" has been assigned to player ", ThisPlayer
ELSE
OUTPUT "No characters with this role are available" ENDIF
ENDPROCEDURE
```

Mark as follows:
MP1 Loop until 'found' or all 45 elements considered
MP2 Test of Player field – i.e. not value **in a loop**
MP3 … AND Role – i.e. match for **ThisRole** parameter **in a loop**
MP4 If available character found, assign ThisPlayer to the character **in a loop**
MP5 When character found set termination condition/flag
MP6 Both OUTPUT messages logically correctly placed
MP7 Both OUTPUT statements correctly formed

(7 marks)

5 Complete the table by giving the appropriate data type in each case.

Variable	Example data value	Data type
Available	TRUE	
Received	"18/04/2021"	
Index	100	

Answer

Variable	Example data value	Data type
Available	TRUE	BOOLEAN
Received	"18/04/2021"	STRING
Index	100	INTEGER

One mark per row

(3 marks)

6 A program is being developed to implement a game for up to six players.

During the game, each player assembles a team of characters. At the start of the game there are 45 characters available.

Each character has four attributes, as follows:

Attribute	Examples	Comment
Player	0, 1, 3	The player the character is assigned to.
Role	Builder, Teacher, Doctor	The job that the character will perform in the game.
Name	Bill, Lee, Farah, Mo	The name of the character. Several characters may perform the same role, but they will each have a unique name.
Skill Level	14, 23, 76	An integer in the range 0 to 100 , inclusive.

The programmer has defined a record type to define each character. The record type definition is shown in pseudocode as follows:

```
TYPE CharacterType
DECLARE Player : INTEGER
DECLARE Role : STRING
DECLARE Name : STRING
DECLARE SkillLevel : INTEGER
ENDTYPE
```

The Player field indicates the player to which the character is assigned (1 to 6). The field value is 0 if the character is **not** assigned to any player.

The programmer has defined a global array to store the character data as follows:

```
DECLARE Character : ARRAY[1:45] OF CharacterType
```

At the start of the game all record fields are initialised, and all Player field values are set to 0

The programmer has defined a program module as follows:

Module	Description

Count()	• called with two parameters:
	  ◦ an integer representing a player
	  ◦ a string representing a character role
	• searches the Character array for characters with the given role that are assigned to the given player
	• counts the number of assigned characters and sums their total skill level
	• outputs the result of the search if characters with the given role are found, for example:
	"Player 3 has 4 characters with the role of Teacher and the total skill level is 65"
	• if no characters with the given role are found, outputs:
	"No characters with that role are assigned to this player"

Complete the pseudocode for module Count().

PROCEDURE Count(ThisPlayer : INTEGER, ThisRole : STRING)

Answer

Mark Scheme and Guidance

Example solution

```
PROCEDURE Count(ThisPlayer : INTEGER, ThisRole : STRING)
DECLARE Index, Num, Total : INTEGER
```

```
Num ← 0
Total ← 0

FOR Index ← 1 TO 45
IF Character[Index].Player = ThisPlayer AND __
Character[Index].Role = ThisRole THEN
Num ← Num + 1
Total ← Total + Character[Index].SkillLevel
NEXT Index

IF Num > 0 THEN
OUTPUT "Player ", ThisPlayer, __
" has ", Num, " characters with the role of
",ThisRole, " and the total skill level is
",Total
ELSE

OUTPUT "No characters with that role are assigned to this player"

ENDIF

ENDPROCEDURE
```

MP1 Initialisation of local integers for Num and Total
MP2 Loop through 45 elements
MP3 Attempt to check Player and Role fields **in a loop**
MP4 Correctly compare Player field with parameter **in a loop**
MP5 Correctly compare Role field with parameter in a loop
MP6 … if player and role found, increment Num **an**d sum Skill Total in a loop
MP7 Test for any matches **after the loop**
MP8 Both possible OUTPUT statements correctly formed following an attempt at MP6 but outputting one only

Max 7 marks

(7 marks)

7 (a) A factory needs a program to help manage its production of items.

Data will be stored about each item.

The data for each item will be held in a record structure of type Component.

The programmer has started to define the fields that will be needed as shown in the table.

Field	Example value	Comment
Item_Num	123478	a numeric value used as an array index
Reject	FALSE	TRUE if this item has been rejected
Stage	'B'	a letter to indicate the stage of production
Limit_1	13.5	any value in the range 0 to 100 inclusive
Limit_2	26.4	any value in the range 0 to 100 inclusive

Write pseudocode to declare the record structure for type Component.

Answer

Mark Scheme and Guidance

Pseudocode:

```
TYPE Component
DECLARE Item_Num : INTEGER
DECLARE Reject : BOOLEAN
DECLARE Stage : CHAR
DECLARE Limit_1 : REAL
DECLARE Limit_2 : REAL
ENDTYPE
```

Mark as follows:

1. One mark for TYPE and ENDTYPE statements
2. One mark for Item_Num and Reject fields

3. One mark for Stage field
4. One mark for Limit fields as REAL

(4 marks)

(b) State **three** benefits of using an array of records to store the data for all items.

Answer

Mark Scheme and Guidance

One mark per point:

1. Allows for iteration / can use a loop **to access the records / data items**
2. Use of index to directly access a **record** in the array // Example of simplification of code e.g. use of dot notation Item[1].Stage
3. Simplifies the code / algorithm // Reduces duplication of code // **Program** easier to write / understand / maintain / test / debug // **Data items/record** easier to search / sort / manipulate

(3 marks)

8 A record structure is declared to hold data relating to components being produced in a factory:

```
TYPE Component
DECLARE Item_ID : STRING
DECLARE Reject : BOOLEAN
DECLARE Weight : REAL
ENDTYPE
```

The factory normally produces a batch (or set) of 1000 components at a time. A global array is declared to store 1000 records for a batch:

```
DECLARE Batch : ARRAY [1:1000] OF Component
```

Two global variables contain the minimum and maximum acceptable weight for each component.
The values represent an inclusive range and are declared as:

```
DECLARE Min, Max : REAL
```

When batches of less than 1000 components are processed, it is necessary to indicate that certain elements in the array are unused.

Suggest how an unused array element could be indicated.

Answer

Mark Scheme and Guidance

One mark for either:

- Set the Item_ID field to an empty string / NULL / invalid value
- Set Weight to <= 0 / zero

(1 mark)

9 Write **pseudocode** statements to declare the record data type FootballClub to hold data about football clubs in a league, to include:

- name of team
- date team joined the league
- main telephone number
- name of the manager
- number of members
- current position in the league.

Answer

Mark Scheme and Guidance

One mark for TYPE FootballClub and ENDTYPE correct
One mark for every two correct declarations

Example answer

```
TYPE FootballClub
DECLARE TeamName : STRING
DECLARE DateOfJoining : DATE
DECLARE MainTelephone : STRING
DECLARE ManagerName : STRING
DECLARE NumberOfMembers : INTEGER
DECLARE LeaguePosition : INTEGER
ENDTYPE
```

(4 marks)

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

The following pseudocode represents part of the algorithm for a program:

```
CASE OF ThisValue
< 30 : Level ← "Low" // less than 30
Check ← 1
< 20 : Level ← "Very Low" // less than 20
Check ← ThisValue / 2
30 TO 40 : Level ← "Medium" // between 30 and 40
Check ← ThisValue / 3
Data[ThisValue] ← Data[ThisValue] + 1
> 40 : Level ← "High"
ENDCASE
```

Give the appropriate data types for the variables ThisValue, Check and Level.

ThisValue .....

Check .....

Level .....

Answer

Mark Scheme and Guidance

One mark per point:

- ThisValue: **INTEGER**
- Check: **REAL**
- Level: **STRING**

(3 marks)

11 (a) Refer to the insert for the list of pseudocode functions and operators

A shop sells car accessories. A customer order is created if an item cannot be supplied from current stock. A program is being developed to create and manage the customer orders.

The following identifier table shows some of the data that will be stored for each order.

Complete the identifier table by adding meaningful variable names and appropriate data types.

Example value	Explanation	Variable name	Data type
"Mr Khan"	The name of the customer		
3	The number of items in the order		
TRUE	To indicate whether this is a new customer		
15.75	The deposit paid by the customer		

Answer

Mark Scheme and Guidance

One mark for each row with appropriate variable name and data type

Example value	Explanation	Variable name	Data type
"Mr Khan"	The name of the customer	**CustomerName**	**STRING**
3	The number of items in the order	**NumItems**	**INTEGER**
TRUE	To indicate whether this is a new customer	**NewCustomer**	**BOOLEAN**
15.75	The deposit paid by the customer	**Deposit**	**REAL**

(4 marks)

(b) The data that needs to be stored for each customer order in part **(a)** is not all of the same type.

Describe an effective way of storing this data for many customer orders while the program is running.

Answer

Mark Scheme and Guidance

One mark per point **Max 3** marks

Declaration
1. Declare a composite / record (type)
2. Declare an array of the given composite / record (type)

Expansion of record:
3. ... containing all data items required // containing items of different data types

Expansion of array:
4. ... where **each array element** represents data for one order / customer (order)

(3 marks)

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

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

The following table shows four valid pseudocode assignment statements.

Complete the table by giving the data type that should be used to declare the variable underlined in each assignment statement.

Assignment statement	Data type
MyVar1 ← Total1 / Total2	
MyVar2 ← 27/10/2023	
MyVar3 ← "Sum1 / Sum2"	
MyVar4 ← Result1 AND Result2	

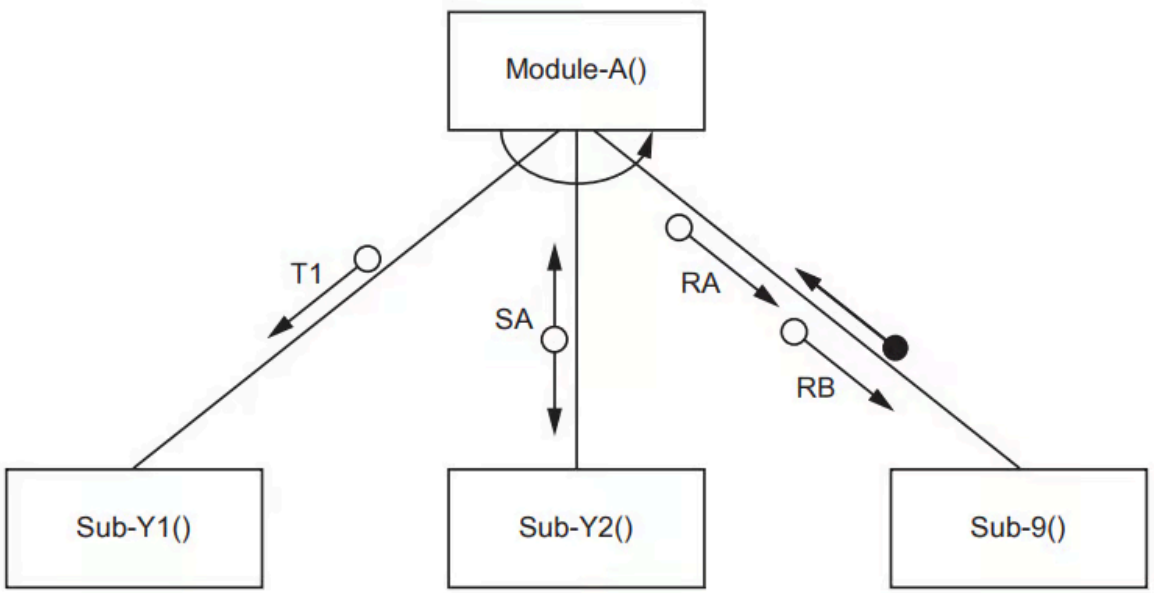
Answer

Mark Scheme and Guidance

Assignment statement	Data type
MyVar1 ← Total1 / Total2	REAL
MyVar2 ← 27/10/2023	DATE
MyVar3 ← "Sum1 / Sum2"	STRING
MyVar4 ← Result1 AND Result2	BOOLEAN

(4 marks)

13 A structure chart shows the modular structure of a program:



The structure chart shows that Sub-9() is a function.

A Boolean value is returned by Sub-9() for processing by Module-A().

The original parameter RA is of type integer and RB is of type string.

A record type MyType will be defined with three fields to store the values passed between the two modules.

Write pseudocode to define MyType.

Answer

Mark Scheme and Guidance

```
TYPE MyType
DECLARE RA : INTEGER
DECLARE RB : STRING
DECLARE RC : BOOLEAN
ENDTYPE
```

MP1 TYPE MyType... ENDTYPE
MP2 RA as integer and RB3 as string
MP3 RC as Boolean

(3 marks)