

3 A program is needed to manage individual rentals in a car-hire business.

The data items for each rental will be held in a record structure of type `RentalRecord`. The programmer has started to define the items that will be needed:

Item	Example value	Comment
RentalID	"AB1234"	a unique alpha-numeric value
CarID	241	a numeric value used as an array index
DisCode	'S'	a letter indicating the type of discount offered
Start	13/06/2025	when the rental starts
Duration	7	the number of days of the rental
Completed	FALSE	TRUE when the car is returned and the rental charge paid

(a) (i) Write pseudocode to declare the record structure for type `RentalRecord`.

[4]

(ii) A 1D array `Rental` containing 500 elements is used to store the data for all rental records.

Write pseudocode to declare the `Rental` array.

..... [2]

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

1

2

3

①

1 (a) The table contains pseudocode examples.

Each example may contain statements that relate to one or more of:

- selection
- iteration (repetition)
- subroutines (procedures or functions).

Complete the table by placing **one or more** ticks (✓) in each row.

Pseudocode example	Selection	Iteration	Subroutine
IF Status = FALSE THEN FOR Count ← 1 TO 20 CALL Reset (Count) NEXT Count ENDIF			
OTHERWISE : Status ← TRUE			
WHILE AllDone() = TRUE			
30 : NextChar ← 'X'			

[4]

(b) Complete the table by giving the appropriate data type.

Variable	Example data value	Data type
Result	5.42	
MonthLetter	"JFMAMJJASOND"	
Birthday	15/11/2009	

[3]

(c) Evaluate each expression in the table by using the data values shown in **(b)**.

Write 'ERROR' if the expression contains an error.

Expression	Evaluates to
INT(Result) + 1 > 6	
NUM_TO_STR(LENGTH(MonthLetter))	
NUM_TO_STR(Result + "3.2")	
MID(MonthLetter, MONTH(Birthday) - 2, 1)	

[4]

1 The composite record data type, ClubMember, is defined in pseudocode as:

```
TYPE ClubMember
  DECLARE Code : INTEGER
  DECLARE LastName : STRING
  DECLARE FirstName : STRING
  DECLARE Telephone : STRING
  DECLARE JoinDate : DATE
  DECLARE Fees : REAL
  DECLARE FeesPaid : BOOLEAN
ENDTYPE
```

(a) (i) Write the **pseudocode** statement to set up a variable for one record of the composite data type, ClubMember.

.....

..... [1]

(ii) Write the **pseudocode** statements to assign the following values to the variable set up in part (a)(i):

- 984632 to Code
- TRUE to FeesPaid

.....

.....

.....

..... [2]

(b) An enumerated data type, Activity, is required, so that a new field, Choice, can be added to the composite data type, ClubMember, to allow members to choose an activity.

(i) Write the **pseudocode** statement for the type declaration of Activity to hold the names of the available activities:

Badminton, Football, Golf, Snooker, Swimming, Tennis.

.....

.....

.....

..... [2]

(ii) Write the **new pseudocode** statement required to update the declaration of Choice in the definition of ClubMember.

.....

..... (3) [1]

- 1 A program is being developed to control the production line in a factory.
- (a) A number of different program life cycles are available for the development of a program.

(i) Explain the need for different program development life cycles.

.....

..... [1]

(ii) Coding is a stage in a program development life cycle.

State **one** consideration that would influence the choice of programming language.

.....

..... [1]
- (b) The program has been in use for a number of months and adaptive maintenance is required.

(i) Give **three** reasons why adaptive maintenance may be required.

1

2

3 [3]

(ii) As well as adaptive maintenance, other types of program maintenance may be needed.

Identify **one** other type of program maintenance.

..... [1]

(c) Pseudocode has been used to design modules for the program to control the production line.

The table shows **four** valid pseudocode expressions.

Complete the table by giving the data type of the evaluated expression.

Expression	Data type
RIGHT (MachineCode, 4)	
Speed * 2.5	
NOT Status	
IS_NUM(Check)	

[4]

(d) A global array `Product` is used as part of the pseudocode design being developed to control the production line. `Product` is used to store the number of rejected items each day.

The following pseudocode statement is used to assign a value to an element of the array:

```
Product[x, y] ← 23
```

The lower and upper bound values are shown in the table:

Variable	Lower bound	Upper bound
x	0	99
y	0	9

(i) State the number of dimensions of `Product`

..... [1]

(ii) State the total number of elements in array `Product`

..... [1]

(iii) Give the pseudocode declaration for the array `Product`

..... [2]

1 A program is being developed for the management of a sports centre.

(a) The programmer developing the software decides to use modules (procedures or functions) and local variables. They also decide not to use global variables.

(i) State **two** reasons why the programmer decides to use modules.

1

.....

2

.....

[2]

(ii) State **one** difference between local and global variables.

.....

..... [1]

(iii) State **two** benefits of using local variables.

1

.....

2

.....

[2]

(b) The pseudocode design contains a number of expressions.

Complete each pseudocode expression so that it evaluates to the value shown.

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

Expression	Evaluates to
..... (68)	' D '
..... (04 / 02 / 2025)	2
..... TRUE	FALSE
..... (..... ("Court1.4Upper" , ,))	1 . 4

[5]

(c) The table lists some of the variables used in the program.

Complete the table by writing the most appropriate data type for each variable.

Variable	Use of variable	Data type
MemberCount	stores how many members have used the sports centre each day	
TotalTakings	stores the total amount of money taken each day	
BookingConfirmed	stores the state of a booking for an exercise class; a booking is either confirmed or not confirmed	
MemberDOB	to calculate when to send an email with a birthday message to a member	

[4]

1 Data types can be defined using pseudocode.

The composite record data type, `Departure`, is used to represent flights from Cambridge Airport and is defined in pseudocode as:

```
TYPE Departure
  DECLARE FlightNumber : STRING
  DECLARE Destination : STRING
  DECLARE FlightDate : DATE
  DECLARE Gate : STRING
  DECLARE Airline : STRING
ENDTYPE
```

A variable, `Flight1`, is declared in pseudocode as:

```
DECLARE Flight1 : Departure
```

(a) Write **pseudocode** to store the following details to `Flight1`:

Field	Data
FlightNumber	SB2789
Destination	Dublin
FlightDate	30/07/2025
Gate	N03
Airline	Cambridge Airways

.....

.....

.....

.....

.....

..... [3]

(b) The data type for `Gate` is changed to an enumerated data type, `GateID`.

(i) Write a **pseudocode** statement to declare `GateID` to hold the identity codes for the airport gates:

N01, N02, N03, W01, W02, W03, W04

.....

..... [2]

(ii) Write the new **pseudocode** statement required to replace the declaration of `Gate` in `Departure`.

1 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
```

(a) The pseudocode contains a logical error.

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

Error

.....

Correction

.....

[2]

(b) 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]

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

Variable name	Data type
HighRate	
TaxPayable	

[2]

(d) The final CASE condition (> 200) in the pseudocode example could be replaced with a keyword.

Give the keyword.

..... [1]

4 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:

1. The two strings represent valid numeric values.
2. The numeric value of string `y` is not zero.

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

Type

.....

Cause

.....

[2]

(b) The designer considers implementing the **checks and calculation** as a module (a procedure or a function). One reason for this is that the same checks and calculations are performed at several places in the program.

Give **another** reason why this is a suitable approach **and** state what is avoided by this approach.

Reason

.....

Avoided

.....

[2]

- (c) 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()`.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

1 (a) The following table contains pseudocode examples.

Each example may contain statements that relate to one or more of the following:

- selection
- iteration (repetition)
- subroutine (procedure or function).

Complete the table by placing **one or more** ticks (✓) in each row.

Pseudocode example	Selection	Iteration	Subroutine
FOR Index ← 1 TO 3 IF Safe[Index] = TRUE THEN Flap[Index] ← 0 ENDIF NEXT Index			
CASE OF Compound(3)			
REPEAT UNTIL AllDone() = TRUE			
WHILE Result[3] <> FALSE			

[4]

(b) 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	

[3]

(c) 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")	

[3]

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

(a) 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"	

[4]

(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	

[4]

(c) The variable names A, B, C and D in part (a) are **not** good programming practice.

(i) State why these variable names are **not** suitable.

[1]

(ii) Identify **one** problem that these variable names might cause.

[1]

(iii) The choice of suitable variable names is one example of good programming practice.

Give **one other** example.

[1]

3 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

(a) (i) Write pseudocode to declare the record structure for type `Component`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(ii) A 1D array `Item` of 2000 elements will store the data for all items.

Write pseudocode to declare the `Item` array.

.....

..... [2]

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

1

.....

2

.....

3

.....