QUESTION 1.

5 Data about sports club members are stored in a random file of records.

- The key field of a member record is the member ID (range 1000 to 9999).
- Other member data are stored.
- A hashing function is used to calculate a record address.
- The random file initially consists of dummy records.
- Dummy records are shown by member ID set to 0.

```
FUNCTION Hash(MemberID : INTEGER) RETURNS INTEGER

   Address ← MemberID MOD 100

   RETURN Address

ENDFUNCTION
```

(a) New members with the following member IDs have joined the sports club:

1001, 3005, 4096, 2098, 7002

Indicate where each record should be stored by deleting the zero and writing the member ID in the correct cell.

MembershipFile

Address	MemberID	Other member data
0	0	
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0	
:		
:		
96	0	
97	0	
98	0	
99	0	

[2]

(b) **(i)** The program stores a new member's data in the record variable `NewMe` `MemberID` stores the member ID.

Complete the pseudocode:

```
10 // generate record address

20 NewAddress ← ..............................................................................

30 // move pointer to the disk address for the record

40 SEEK ........................................................................................

50 PUTRECORD "MembershipFile", ..........................................................
```

[4]

(ii) Before records can be saved to the file `MembershipFile`, the file needs to be opened.

Complete the pseudocode.

```
01 TRY

02     OPENFILE ................................................................ FOR RANDOM

03 EXCEPT

04     ...........................................................................................

05 ENDTRY
```

[2]

(iii) A record with member ID 9001 is to be stored.

Explain the problem that occurs when this record is saved.

....

....

....

....[2]

(iv) Describe a method, without changing the function `Hash`, to handle the problem identified in **part (b)(iii)**.

....

....

....

....[2]

(v) Write **pseudocode** to implement the method you described in **part (b)(**

Choose line numbers to indicate where your pseudocode should be inse

QUESTION 2.

6 A company keeps details of its stock items in a file of records, `StockFile`.

(a) The record fields are the `ProductCode`, the `Price` and the `NumberInStock`.

Write the **program code** to declare the record structure `StockItem`.

Programming language ....

....

....

....

....

....

....

.... [4]

(b) Before records can be read from file `StockFile`, the file needs to be opened.

(i) Complete the pseudocode.

```
01  TRY
02      OPENFILE ..............................................................................
03  EXCEPT
04      ..................................................................................
05  ENDTRY
```

[2]

(ii) Explain the reason for including lines `01`, `03`, `04`, `05`.

....

....

....

.... [2]

(c) A stock report program uses a variable of type `StockItem` declared as follo

```
DECLARE ThisStockItem : Stockitem
```

The program reads each record in the file `StockFile` in turn.

The program outputs the fields `ProductCode` and `NumberInStock` for each record.

Write **pseudocode** for this.

....

....

....

....

....

....

....

.... [4]

BLANK PAGE

QUESTION 3.

5 Data about sports club members are stored in a random file of records.

- The key field of a member record is the member ID (range 1000 to 9999).
- Other member data are stored.
- A hashing function is used to calculate a record address.
- The random file initially consists of dummy records.
- Dummy records are shown by member ID set to 0.

```
FUNCTION Hash(MemberID : INTEGER) RETURNS INTEGER

   Address ← MemberID MOD 100

   RETURN Address

ENDFUNCTION
```

(a) New members with the following member IDs have joined the sports club:

1001, 3005, 4096, 2098, 7002

Indicate where each record should be stored by deleting the zero and writing the member ID in the correct cell.

MembershipFile

Address	MemberID	Other member data
0	0	
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0	
:		
:		
96	0	
97	0	
98	0	
99	0	

[2]

(b) (i) The program stores a new member's data in the record variable `NewM` `MemberID` stores the member ID.

Complete the pseudocode:

```
10 // generate record address

20 NewAddress ← ........................................................................

30 // move pointer to the disk address for the record

40 SEEK ..................................................................................

50 PUTRECORD "MembershipFile", ....................................................
```

[4]

(ii) Before records can be saved to the file `MembershipFile`, the file needs to be opened.

Complete the pseudocode.

```
01 TRY

02    OPENFILE ................................................. FOR RANDOM

03 EXCEPT

04    ....................................................................................

05 ENDTRY
```

[2]

(iii) A record with member ID 9001 is to be stored.

Explain the problem that occurs when this record is saved.

....

....

....

....[2]

(iv) Describe a method, without changing the function `Hash`, to handle the problem identified in **part (b)(iii)**.

....

....

....

....[2]

(v) Write **pseudocode** to implement the method you described in **part (b)(**

Choose line numbers to indicate where your pseudocode should be inse

QUESTION 4.

3 A programmer is writing a treasure island game to be played on the compute a rectangular grid, 30 squares by 10 squares. Each square of the island is repre element in a 2D array. The top left square of the island is represented by the array ele There are 30 squares across and 10 squares down.

The computer will:

- generate three random locations where treasure will be buried
- prompt the player for the location of one square where the player chooses to dig
- display the contents of the array by outputting for each square:
  - `'.'` for only sand in this square
  - `'T'` for treasure still hidden in sand
  - `'X'` for a hole dug where treasure was found
  - `'O'` for a hole dug where no treasure was found.

Here is an example display after the player has chosen to dig at location [9, 3]:

```
..............................
..............................
..............................
..............................
..............................
........T.....................
..............................
..............................
.........T....................
...X..........................
```

The game is to be implemented using object-oriented programming.

The programmer has designed the class `IslandClass`. The identifier table for this class is:

Identifier	Data type	Description
`Grid`	`ARRAY[0 : 9, 0 : 29] OF CHAR`	2D array to represent the squares of the island
`Constructor()`		instantiates an object of class `IslandClass` and initialises all squares to sand
`HideTreasure()`		generates a pair of random numbers used as the grid location of treasure and marks the square with `'T'`
`DigHole(Row, Column)`		takes as parameters a valid grid location and marks the square with `'X'` or `'O'` as appropriate
`GetSquare(Row, Column)`	`CHAR`	takes as parameter a valid grid location and returns the grid value for that square from the `IslandClass` object

(a) The programmer designed the pseudocode for the main program as follows:

```
DECLARE Island : IslandClass.Constructor()      // instantiat
CALL DisplayGrid()                            // output island squ
FOR Treasure ← 1 TO 3                             // hide 3 treasures
   CALL Island.HideTreasure()
ENDFOR
CALL StartDig()                       // user to input location of dig
CALL DisplayGrid()                            // output island squares
```

Write **program code** to implement this pseudocode.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....[3]

(b) Write **program code** to declare the `IslandClass` and write the constructor

The value to represent sand should be declared as a constant.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(c) The procedure `DisplayGrid` shows the current grid data. `DisplayGrid` getter method `GetSquare` of the `Island` class.

An example output is:

```
..............................
..............................
..............................
..............................
..............................
........T.....................
..............................
..............................
.........T....................
...X..........................
```

(i) Write **program code** for the `GetSquare(Row, Column)` getter method.

[2]

(ii) Write **program code** for the `DisplayGrid` procedure.

[4]

(d) Write **program code** for the `HideTreasure` method. Your method shoul
random location generated does not already contain treasure.

The value to represent treasure should be declared as a constant.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(e) **(i)** The `DigHole` method takes two integers as parameters. These parame grid location. The location is marked with `'X'` or `'O'` as appropriate.

Write **program code** for the `DigHole` method. The values to represent treasu treasure and hole should be declared as constants.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

.... [3]

(ii) The `StartDig` procedure:

- prompts the player for a location to dig
- validates the user input
- calls the `DigHole` method from **part (e)(i)**.

Write **program code** for the `StartDig` procedure. Ensure that the user input is fully validated.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(f) **(i)** The squares in the `IslandClass` grid could have been declared
`Square` class.

State the term used to describe the relationship between `IslandClass` and `Sq`

....

....[1]

(ii) Draw the appropriate diagram to represent this relationship. Do not list the attributes and methods of the classes.

[2]

BLANK PAGE

BLANK PAGE

QUESTION 5.

..

6 A programmer wants to create a computer simulation of animals searching for
The desert is represented by a 40 by 40 grid. Each position in the grid is represente
coordinates. `'A'` represents an animal and `'F'` represents food. At the start of the sim
grid contains 5 animals and 1 food source.

The following is an example of part of the grid.

	0	1	2	3	4	...	37	38	39
0	A					..			
1			F			..			
2						..		A	
3				A		..			
...	..	..	..	..	..	..	..	..	..
38				A		..	A		
39						..			

A timer is used. In each time interval, each animal randomly moves 0 or 1 position in a random direction. The program generates this movement by computing two random numbers, each of which can be –1, 0 or 1. The program adds the first random number to the across number and the second random number to the down number representing the animal's position.

For example:

- if 0 and 1 are generated, the across value does not change, the down value increases by 1
- if –1 and 1 are generated, the across value decreases by 1, and the down value increases by 1.

Each animal has an individual score. If the animal moves to a position in the grid with food (`'F'`):

- the animal's score increases by 1
- the food disappears
- one new animal (`'A'`) is randomly generated and added to the grid (to a maximum of 20 animals)
- one new food (`'F'`) is randomly generated and added to the grid.

The simulation is to be implemented using object-oriented programming.

The programmer has designed two classes, `Desert` and `Animal`.

The `Desert` class consists of:

- attributes
  - `Grid`
  - `StepCounter`
  - `AnimalList`
  - `NumberOfAnimals`
- methods
  - `Constructor`
  - `IncrementStepCounter`
  - `GenerateFood`
  - `DisplayGrid`

Each attribute consists of a value and a get and set method that allow access to the attributes.

The following table describes the attributes and methods for the `Animal` class.

Identifier	Data type	Description
`Constructor()`		Instantiate an object of the `Animal` class • Generate a pair of random numbers between 0 and 39. • Place animal at that random position. • Initialise the animal's score to 0.
`EatFood()`		• Delete the food. • Increase the score of the animal that called the method. • Call the `GenerateFood` method of the `Desert` class. • Call the `Constructor` method of the `Animal` class.
`Move()`		• Call the `GenerateChangeInCoordinate` method for each coordinate (across or down number) of the animal's position. • Moves the animal to the new space. • If there is food in the new position, call the `EatFood` method.
`Score`	`INTEGER`	Initialised to 0
`Across`	`INTEGER`	The across value, between 0 and 39
`Down`	`INTEGER`	The down value, between 0 and 39

(a) Write **program code** to declare the attributes and constructor for the Anima

You only need to write the set and get methods for the attribute `Across`.

You should also write:

- the constructor for the class
- set and get methods for the `Across` attribute only.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

(b) The `Constructor` method of the `Desert` class:

- initialises an empty grid
- creates 5 animal objects which are added to the `AnimalList` (an array of anima
currently on the grid)
- generates one food
- sets the `StepCounter` to 0.

Write **program code** for the `Constructor` method.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(c) **(i)** The function `GenerateChangeInCoordinate`:

- receives a coordinate (across or down number) as a parameter
- checks whether the coordinate's value is at a boundary of the grid
- returns a random change (–1, 0 or 1) that will keep the animal's position with grid.

Write **program code** for the `GenerateChangeInCoordinate` function.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....[4]

QUESTION 6.

2 Kendra collects books. She is writing a program to store and analyse information

Her program stores information about each book as a record. The following table
information that will be stored about each book.

Field name	**Description**
`Title`	The title of the book
`Author`	The first listed author of the book
`ISBN`	A 13-digit code that uniquely identifies the book, for example: `"0081107546738"`
`Fiction`	If the book is fiction (`TRUE`) or non-fiction (`FALSE`)
`LastRead`	The date when Kendra last read the book

(a) Write **pseudocode** to declare an Abstract Data Type (ADT) named `Book`, to store the information in the table.

....

....

....

....

....

....

....

....

.... [4]

(b) The records are stored in a random access file.

The function, `Hash()`, takes as a parameter the ISBN and returns the hash value.

The disk address of the record in the hash table is calculated as: ISBN modulus 2000 p

Write **program code** for the function `Hash()`.

Programming language ....

Program code ....

....

....

....

....

....

....

.... [4]

(c) The random access file, `MyBooks.dat`, stores the data about the books in t

```
<Title>
<Author>
<ISBN>
<Fiction>
<LastRead>
```

A procedure, `FindBook()`:

- prompts the user to input the ISBN of a book until the ISBN contains 13 numeric digits
- uses the function `Hash()` to calculate the disk address of the record
- reads the record for that book from `MyBooks.dat` into a variable of type `Book`
- outputs all the data about the book.

Use **pseudocode** to write the procedure `FindBook()`.

You can assume that the record exists at the disk address generated.

[8]