# QUESTION 1.

**2** A program displays a menu with choices 1 to 4. The code to display the menu procedure `DisplayMenu`.

**(a)** Pseudocode which uses this procedure is:

```
CALL DisplayMenu
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4
```

**(i)** Describe what this pseudocode will do.

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

.......................................................................................................................................[3]

**(ii)** State why a loop is required.

..........................................................................................................................................

.......................................................................................................................................[1]

**(b)** The following pseudocode is a revised design.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

**(i)** Give the maximum number of inputs the user could be prompted to make.

.............................................. [1]

**(ii)** State why this algorithm is an improvement on the one given in **part (a)**.

..........................................................................................................................................

.......................................................................................................................................[1]

**(c)** The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following e
choice.

| Menu choice | Description | Program response |
|---|---|---|
| 1 | Read data from the customer file | Calls a procedure `ReadFile` which for testing purposes outputs the message “Read file code” |
| 2 | Add a customer | Outputs message “Add  customer code” |
| 3 | Search for a customer | Outputs message “Search customer code” |
| 4 | Terminates the program | Ends |

Complete the pseudocode for the design in **part (b)**, shown again below, to respond to each menu choice.

```
CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT
   OUTPUT "Enter choice (1..4)"
   INPUT Choice
   NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i
```

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

.......................................................................................................................................[3]

**(d)** The algorithm in **part (c)** is to be amended. The program will:

- repeatedly display the menu and respond to the user's choice
- terminate when the user enters 4

Write **program code** for this final design which will be made up of:

- the main program
- procedure `ReadFile`
- procedure `DisplayMenu`

*Visual Basic and Pascal: You should include the declaration statements for variables.*
*Python: You should show a comment statement for each variable used with its data type.*

Programming language ..........................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

.....................................................................................................................................[8]

# QUESTION 2.

**4** A programming language has the built-in function `CONCAT` defined as follows:

```
CONCAT(String1 : STRING, String2 : STRING [, String3 : STRING] )  RETURNS
```

For example: `CONCAT("San", "Francisco") returns "SanFrancisco"`
`CONCAT("New", "York", "City") returns "NewYorkCity"`

The use of the square brackets indicates that the parameter is optional.

**(a)** State the value returned by the following expressions.

If the expression is not properly formed, write ERROR.

**(i)** `CONCAT("Studio", 54)` .................................................... [1]

**(ii)** `CONCAT("parity", "error", "check")` .................................................... [1]

**(iii)** `CONCAT(CONCAT("Binary", "▼", "Coded"), "▼", "Decimal")`

▼ indicates a <Space> character

.............................................................................................................................[2]

**(b)** A country has a number of banks. There are cash dispensers all over the country. Each bank is responsible for a number of dispensers.

- banks have a three digit code in the range 001 – 999
- each dispenser has a five digit code in the range 00001 – 99999

A text file, `DISPENSERS`, is to be created.

It has one line of text for each dispenser. For example: `00342▼007`.

This line in the file is the data for dispenser `00342` which belongs to bank `007`.

Incomplete pseudocode follows for the creation of the file `DISPENSERS`.

For the creation of the file, data is entered by the user at the keyboard.

**(i)** Complete the **pseudocode**.

```
OPENFILE .................................................. FOR WRITE

...................................................................................................................................

   OUTPUT "Enter dispenser code (XXXXX to end)"
   INPUT DispenserCode
   IF DispenserCode <> "XXXXX"
      THEN
         OUTPUT "Enter bank code"
         INPUT BankCode
         LineString ← CONCAT(.........................................,"▼", BankCode)
         // now write the new line to the file

         ...................................................................................................

   ENDIF

UNTIL ...........................................................................................................

...................................................................................................................................

OUTPUT "DISPENSERS file now created"
```

[6]

**(ii)** No attempt has been made to validate the data entered by the user.

Describe **two** different types of validation check for the data entry.

1 .............................................................................................................................

.................................................................................................................................

2 .............................................................................................................................

.................................................................................................................................[2]

**(iii)** The programmer coded this algorithm above and the user successfully entered 15 dispenser records into the text file.

There is data for another 546 dispensers which needs to be added.

State the error that will occur if the user runs the program a second time for further data entry.

.................................................................................................................................[1]

**(iv)** Give the 'file mode' available in the programming language which will be used to address this issue.

.................................................................................................................................[1]

**(c)** The complete data file is created with the structure shown.

A new program is to be written to search the file.

The program will:

- input a bank code
- output a list of all the dispensers which belong to this bank
- output the total number of dispensers for this bank

```
00001▼007
00002▼001
00003▼002
00004▼003
00005▼101
00006▼004
00007▼004

00024▼002
00025▼003
00026▼007
00027▼007
00028▼102

99867▼013
```

An example of a run of the program is shown:

```
Enter bank code 007
00001
00011
00022
00026
00027

There are 5 dispensers for this bank
```

Write the **program code.** Do not attempt to include any validation checks.

*Visual Basic and Pascal: You should include the declaration statements for varia*

*Python: You should show a comment statement for each variable used with its data t*

Programming language ...............................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

............................................................................................................................[10]

# QUESTION 3.

**7** ASCII character codes are used to represent a single character.

Part of the code table is shown below.

**ASCII code table (part)**

| Character | Decimal | Character | Decimal | Character | Decimal |
|---|---|---|---|---|---|
| <Space> | 32 | I | 73 | R | 82 |
| A | 65 | J | 74 | S | 83 |
| B | 66 | K | 75 | T | 84 |
| C | 67 | L | 76 | U | 85 |
| D | 68 | M | 77 | V | 86 |
| E | 69 | N | 78 | W | 87 |
| F | 70 | O | 79 | X | 88 |
| G | 71 | P | 80 | Y | 89 |
| H | 72 | Q | 81 | Z | 90 |

Some pseudocode statements follow which use these built-in functions:

`CHARACTERCOUNT(ThisString : STRING) RETURNS INTEGER`
returns the number of characters in the string `ThisString`.
For example: `CHARACTERCOUNT("South Africa")` returns 12.

`CHR(ThisInteger : INTEGER) RETURNS CHAR`
returns the character with ASCII value `ThisInteger`.
For example: `CHR(66)` returns 'B'.

`ASC(ThisCharacter : CHAR) RETURNS INTEGER`
returns the ASCII value for character `ThisCharacter`.
For example: `ASC('B')` returns 66.

**(a)** Give the values assigned to the variables `A`, `B`, `C` and `D`.

The & operator is used to concatenate two strings.

The expression could generate an error; if so, write ERROR.

| | | |
|---|---|---|
| `Num1 ← 5` | | |
| `A ← ASC('F') + Num1 + ASC('Z')` | **(i)** `A` ............................................................ | [1] |
| `B ← CHR(89) & CHR(69) & CHR(83)` | **(ii)** `B` ............................................................ | [1] |
| `C ← CHARACTERCOUNT(B & "PLEASE")` | **(iii)** `C` ............................................................ | [1] |
| `D ← ASC(ONECHAR("CURRY SAUCE", 7))` | **(iv)** `D` ............................................................ | [1] |

**(b)** A program is to be written to input a message string and then encrypt the me

Study the following pseudocode:

# QUESTION 4.

**5** Toni has a large collection of jazz CDs that are stored in different places. She where the CDs are stored. She decides to write a program to do this.

The program must store the data in a file, `MyMusic`.

**(a)** **(i)** Why is a file needed?

..................................................................................................................................

..................................................................................................................................[1]

**(ii)** `MyMusic` is a text file with the data for each CD as one line of text.

Data for a typical CD are:

| | |
|---|---|
| Title: | Kind of Green |
| Artist: | Miles Coltrane |
| Location: | Rack1-5 |

The line will be formed by concatenating the three data items.

For the example above, the line stored will be:

```
Kind of GreenMiles ColtraneRack1-5
```

Describe a problem that might occur when organising the data in this way.

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

Describe a possible solution.

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

[4]

**(b)** Toni must input the data into the file for all of her CDs.

A procedure, `InputData`, is needed to do this.

Toni designs the procedure and chooses the following identifiers:

| **Identifier** | **Data type** |
|---|---|
| `CDTitle` | `STRING` |
| `CDArtist` | `STRING` |
| `CDLocation` | `STRING` |

The procedure repeatedly performs the following steps:

- input a CD title (A rogue value of "##" is to be used to end the input)
- input the artist
- input the location
- create the text line
- write the text line to the file

When the rogue value is encountered the file is closed.

Write **program code** for the procedure `InputData`.

*Visual Basic and Pascal: You should include declaration statements for variables.*
*Python: You should show a comment statement for each variable used with its data ty*

Programming language ..........................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

.........................................................................................................................[8]

# QUESTION 5.

**5** Claudia stores her large collection of music CDs in different places. Claudia wants she stores each CD. She decides to write a program to do this.

Data items for a typical CD are:

| | |
|---|---|
| Title: | Kind of Green |
| Artist: | Miles Coltrane |
| Location: | Rack3-23 |

The data is to be stored in a text file, `MyMusic`. Each line of the text file will be a string, formed by concatenating the three data items.

Before concatenation, the title and artist will each be made into a fixed-length string of 40 characters. Space characters may need to be added to each data item.

The location is always 8 characters long.

**(a)** **(i)** Explain the benefit of making the stored data into fixed-length strings.

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

...........................................................................................................................

State a drawback of this file design.

...........................................................................................................................

...........................................................................................................................

[3]

**(ii)** When Claudia buys a new CD, the CD data must be added to the existi She has written a procedure in pseudocode. This has the following statements:

```
OPENFILE "MyMusic" FOR WRITE
WRITEFILE "MyMusic", OutputString
CLOSEFILE "MyMusic"
```

There is a problem with the logic of this pseudocode.

State the problem.

.............................................................................................................................

.............................................................................................................................

Identify the effect it will have if the final code is implemented in this way.

.............................................................................................................................

.............................................................................................................................

Give a possible solution.

.............................................................................................................................

............................................................................................................................. [3]

**(b)** Claudia needs to output a list of all the CDs in a particular location.

She designs a procedure, `OutputLocationList`, to do this. She also chooses the following identifiers:

| Identifier | Data type |
|---|---|
| `CDTitle` | `STRING` |
| `CDArtist` | `STRING` |
| `CDLocation` | `STRING` |

The procedure will:

- prompt for the name of the location
- input the location (such as “Rack3-23”)
- search the file for all CDs at this location
- output the title and artist of each CD found
- output the total number of CDs found at that location (such as “17 CDs found”)

Write **program code** for the procedure `OutputLocationList`.

*Visual Basic and Pascal: You should include the declaration statements for variabl*
*Python: You should show a comment statement for each variable used with its data ty*

Programming language ..........................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

.......................................................................................................................... [10]

# QUESTION 6.

**5** Study the following pseudocode statements.

```
CONST  Pi =  3.1           : REAL
DECLARE Triangle, Base, Height, Radius, Cone : REAL
DECLARE a, b, c, Answer2 : INTEGER
DECLARE Answer1          : BOOLEAN
Base ← 2.6
Height ← 10
Triangle ← (Base * Height) / 2
Radius ← 1
Height ← 2
Cone  ← 2 * Pi * Radius * (Radius + Height)
a ← 13
b ← 7
c ← 3

Answer1 ← NOT((a + b + c) > 28)
Total ← 34
Total ← Total - 2
Answer2 ← a + c * c
```

Give the final value assigned to each variable.

**(i)** `Triangle` .................................. [1]

**(ii)** `Cone` .................................. [1]

**(iii)** `Answer1` .................................. [1]

**(iv)** `Total` .................................. [1]

**(v)** `Answer2` .................................. [1]

**(e)** A first attempt was made at writing the 'Search for product code' module.
Ahmed designs this as a function `ProductCodeSearch`.

The function returns an integer value as follows:

- if the product code is found, it returns the index position of the 1D array `PCode` being searched
- if the product code is not found, the function returns -1

Write **program code** for function `ProductCodeSearch`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ...............................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

.............................................................................................................................[6]

# QUESTION 7.

**6** Study the sequence of pseudocode statements.

```
CONST   a = 3.2    : REAL
DECLARE x, y, z, Answer1, Answer2, Answer3  : REAL
DECLARE p, q       : BOOLEAN
x ← 3
x ← x + 7
y ← 6
Answer1 ← 2 * (a + y)
z ← 6
Answer2 ← y ^ 2 + 5
p ← TRUE
q ← NOT(NOT(p))
Answer3 ← y + a * 2
```

Give the final value assigned to each variable.

**(i)** `x` ............................................ [1]

**(ii)** `Answer1` ............................................ [1]

**(iii)** `Answer2` ............................................ [1]

**(iv)** `q` ............................................ [1]

**(v)** `Answer3` ............................................ [1]

**(ii)** Consider the following two statements.

Write either TRUE **or** FALSE next to each statement.

| **Statement** | **TRUE or FALSE** |
|---|---|
| The pseudocode considers all the scores for a player, before progressing to the next player. | |
| The pseudocode considers all scores in a game, before progressing to the next game. | |

[1]

**(iii)** The programmer has made logic errors in the design.

State a line number at which an error occurs.

Explain the error or write the corrected pseudocode statement.

Line number ..............................

Explanation ....................................................................................................................

.............................................................................................................................. [1]

QUESTION 8.

**6** A multi-user computer system makes use of passwords.

To be valid, a password must comply with the following rules:

- at least two lower-case alphabetic characters
- at least two upper-case alphabetic characters
- at least three numeric characters
- alpha-numeric characters only

A function, `ValidatePassword`, is needed to check that a given password follows these rules. This function takes a string, `Pass`, as a parameter and returns a Boolean value:

- `TRUE` if `Pass` contains a valid password
- `FALSE` otherwise.

**(a)** Write **program code** to implement the new function `ValidatePassword`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ..........................................................................................

Program code

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

.......................................................................................................................

Write **program code** for the procedure `SearchFile`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ............................................................................................................

Program code

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

.....................................................................................................................................[10]

QUESTION 9.

**6** A computerised vehicle licensing system stores details about vehicles and their re
(number plates or license plates).

To be valid, a vehicle registration must comply with the following rules:

- It must be between six and nine characters long.
- Characters 1 to 3 are upper case alphabetic characters.
- Characters 4 to 5 are numeric characters.
- Remaining characters are upper case alphabetic.

A function, `ValidateRegistration` is needed to check that a given registration mark follows these rules. This function takes a string, `Registration` as a parameter and returns a Boolean value:

- `TRUE` if it is a valid registration
- `FALSE` otherwise.

**(a)** Write **program code** to implement the new function, `ValidateRegistration`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ..............................................................................................

Program code

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

**(b)** A procedure, `LogEvents`, is required to add the log entry data from `LogArray` to the existing text file, `LoginFile.txt`.

Unused array elements are assigned the value `"****"`. These can occur anywhere in the array and should not be written to the file.

Write **program code** for the procedure `LogEvents`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ...................................................................................................

Program code

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...................................................................................................................................... [8]

# QUESTION 10.

**3** **(a)** A multi-user computer system stores information about users. It use `UserNameArray`, of type `STRING`. There are 100 elements in the array.

The format of the string in each element of the array is as follows:

```
<UserID><UserName>
```

- `UserID` is a six-character string of numerals.
- `UserName` is a variable-length string.

Write **pseudocode** for a procedure, `BubbleSort`, to perform an efficient bubble sort on `UserNameArray`. The array is to be sorted in ascending order of `UserID`.

You should assume that `UserNameArray` has been declared as a global variable.

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................[8]

**(b)** The value of `UserID` should be unique for each user but a problem ha repeated `UserID` values may have been issued.

The array is sorted by `UserID`, so any repeated `UserID` values will appear in co array elements.

A procedure, `FindRepeats` is required.

This will:

- compare each element with the previous element and output the `UserID` and `UserName` if the `UserID` is repeated
- output the total number of `UserID`s that are repeated.

For example, the `UserNameArray` contains the following entries.

| **Array element** | **Comment** |
|---|---|
| ≀ | |
| `122222Jim Moriarty` | |
| ≀ | |
| `123456Fred Smith` | |
| `123456Eric Sykes` | Repeated User ID |
| `123456Kevin Turvey` | Repeated User ID |
| ≀ | |
| `222244Alice Chan` | |
| `222244Myra Singh` | Repeated User ID |
| ≀ | |
| `333333Yasmin Halim` | |
| ≀ | |

For this example, the output is:

```
123456Eric Sykes
123456Kevin Turvey
222244Myra Singh
There are 3 repeated UserIDs
```

If no repeated `UserID`s are found, the output is:

```
The array contains no repeated UserIDs
```

Write **program code** for the procedure, `FindRepeats`.
You should assume that `UserNameArray` has been declared as a global varia

Visual Basic and Pascal: You should include the declaration statements for variables
Python: You should show a comment statement for each variable used with its data typ

Programming language ..............................................................................................

Program code

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

.............................................................................................................................[8]

**(c)** **(i)** The `FindRepeats` procedure forms part of a program.

Name **three** stages in a program development cycle.

1 ..........................................................................................................................................

2 ..........................................................................................................................................

3 ..........................................................................................................................................
[3]

**(ii)** The program containing `FindRepeats` will be created using an IDE.

State what is meant by IDE.

............................................................................................................................................

............................................................................................................................................[1]

**(iii)** Name **two** features provided by an IDE that assist in the program development cycle.

1 ..........................................................................................................................................

............................................................................................................................................

2 ..........................................................................................................................................

............................................................................................................................................
[2]

**(iv)** The procedure, `FindRepeats`, is written assuming there are 100 elements in `UserNameArray`.

In the main program, the global array, `UserNameArray`, has been declared with only 50 elements.

State the type of error this will cause.

............................................................................................................................................[1]

# QUESTION 11.

**4** Programming languages provide built-in functions to generate random numbers.
To be truly random, the frequency of each number generated should be the same.

You are required to write program code to test the random number generator of your language.

The test should:

- generate a given number of random numbers between 1 and 10 inclusive
- keep a count of the number of times each number is generated
- calculate the expected frequency of each number 1 to 10
- output the actual frequency of each number 1 to 10
- output the difference between the actual frequency and the expected frequency.

The program code should be written as a procedure. In pseudocode, the procedure heading will be:

```
PROCEDURE TestRandom(Repetitions AS INTEGER)
```

The parameter, `Repetitions`, contains a value representing the total number of random numbers that should be generated.

The following example shows the expected output for the procedure call, `TestRandom(200)`.

```
    The expected frequency is 20.

Number     Frequency     Difference
  1            17             -3
  2            21              1
  3            12             -8
  4            28              8
  5            20              0
  6            19             -1
  7            21              1
  8            16             -4
  9            24              4
  10           22              2
```

**(a)** Write **program code** for the procedure, `TestRandom`.

Visual Basic and Pascal: You should include the declaration statements for variab
Python: You should show a comment statement for each variable used with its data t

Programming language ..........................................................................................

Program code

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

...............................................................................................................................

........................................................................................................................ [16]

**(b)** Name **three** features of a typical IDE that would help a programmer to debug

Explain how each of these could be used in the debugging of the `TestRandom`
from **part (a)**.

Feature 1 ..............................................................................................................

Explanation ..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

Feature 2 ..............................................................................................................

Explanation ..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

Feature 3 ..............................................................................................................

Explanation ..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................
[6]

**(c)** The procedure is developed and run using the call `TestRandom(200)`. No system errors are produced.

To ensure that the procedure works correctly, you need to check the output.

Describe **two** checks you should make to suggest the program works correctly.

1 ..............................................................................................................

..............................................................................................................

..............................................................................................................

2 ..............................................................................................................

..............................................................................................................

..............................................................................................................
[2]

# QUESTION 12.

**4** Part of a program written in pseudocode is shown.

```
01  DECLARE NumElements : INTEGER

…

10  FUNCTION ScanArray(SearchString : STRING) RETURNS INTEGER
11
12     DECLARE ArrayIndex : INTEGER
13     DECLARE ArrayString : STRING
14     DECLARE NumberFound : INTEGER
15
16     ArrayIndex ← 0
17     NumberFound ← 0
18
19     FOR ArrayIndex ← 1 TO NumElements
20        ArrayString ← ResultArray[ArrayIndex, 1]
21        IF ArrayString = SearchString
22           THEN
23              CALL SaveToFile(ArrayString)
24              NumberFound ← NumberFound + 1
25        ENDIF
26     ENDFOR
27
28     RETURN NumberFound
29
30  ENDFUNCTION
```

**(a)** **(i)** Examine the pseudocode **and** complete the following table.

| | **Answer** |
|---|---|
| The identifier name of a global integer | |
| The identifier name of a user-defined procedure | |
| The line number of an unnecessary statement | |
| The scope of `ArrayString` | |

[4]

**(ii)** Describe in detail the purpose of lines `19` to `26` in the function `ScanArray()`.
Do **not** use pseudocode in your answer.

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................[4]

**(b)** The function `ScanArray()` needs to be amended so that the compari sensitive. For example, comparing “`Aaaa`” with “`AAAa`” should evaluate to `TRUE`

Write **program code** to implement the **amended** `ScanArray()` function.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ...........................................................................................................

Program code

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................[6]

**(c)** The function `ScanArray()` is one of a number of sub-tasks within a progra

Name the process that involves the splitting of a problem into sub-tasks **and** advantages of this approach.

Name ..............................................................................................................................

Advantage 1 ...................................................................................................................

.........................................................................................................................................

Advantage 2 ...................................................................................................................

.........................................................................................................................................

[3]

**(d)** `ResultArray` is a 2D array of type `STRING`. It represents a table containing 100 rows and 2 columns.

Write **program code** to declare `ResultArray` **and** set all elements to the value `'*'`.

Programming language .......................................................................................................

Program code

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................

.........................................................................................................................................[3]

Question 5 begins on the next page.

# QUESTION 13.

**6** Nadine is developing a program to store the ID and preferred name for each stud
For example, student Pradeep uses the preferred name “Prad”.

The program will:

1. prompt and input a valid user ID and a preferred name
2. write the user ID and preferred name to one of two files
3. allow the user to end the program or repeat from step 1.

The program will consist of three separate modules. Each module will be implemented using either a procedure or a function.

Nadine has defined the modules as follows:

| Module | Description |
|---|---|
| `TopLevel()` | • Call `GetInfo()` to obtain a string containing a valid user ID and a preferred name<br>• Call `WriteInfo()` to write the string to either `File1.txt` or `File2.txt` depending on the first character of the user ID as follows:<br>○ ‘A’ to ‘M’: writes to `File1.txt`<br>○ ‘N’ to ‘Z’: writes to `File2.txt`<br>For example, a string with a user ID of `"G1234"` writes to `File1.txt`<br>• End the program if the file write was unsuccessful<br>• Input (Y/N) to either repeat for the next user ID or to end the program |
| `GetInfo()` | • Input a user ID and repeat until the user ID is valid<br>• Input a preferred name. This will be an empty string if no preferred name is input.<br>• Concatenate the user ID and preferred name using a `'*'` character as a separator and return this string |
| `WriteInfo()` | • Open the file<br>• Append the concatenated string to the file<br>• Close the file<br>• Return a Boolean value:<br>○ `TRUE` if the file write was successful<br>○ `FALSE` if the file write failed, for example, if the disk was full |

A valid user ID:

- is five characters in length
- has a single upper case alphabetic character followed by four numeric characters, for example “G1234”.

Nadine has decided that global variables and nested modules must not be used.

Nadine wants all inputs to have suitable prompts.

**(a)** Write **program code** for the module `GetInfo()`.

Visual Basic and Pascal: You should include the declaration statements for variab
Python: You should show a comment statement for each variable used with its data t

Programming language ..............................................................................................

Program code

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

................................................................................................................................. [8]

**(b)** Write **program code** for the module `TopLevel()`.

Visual Basic and Pascal: You should include the declaration statements for variab
Python: You should show a comment statement for each variable used with its data t

Programming language ..........................................................................................

Program code

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

.............................................................................................................................

....................................................................................................................... [8]

**(c)** Write **pseudocode** for the module declaration of `WriteInfo()`.

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

.............................................................................................................................................. [3]

# QUESTION 14.

**6** A text file, `StudentContact.txt`, contains a list of names and telephone num in a school. Not all students in the school have provided a contact telephone numbe their name will not be in the file.

Each line of the file is stored as a string that contains a name and telephone number, separat the asterisk character (`'*'`) as follows:

`<Name>'*'<TelNumber>`, for example:

`"Bill Smith*081234567"`

A 1D array, `ClassList`, contains the names of students in a particular class. The array consists of 40 elements of string data type. You can assume that student names are unique. Unused elements contain the empty string `""`.

A program is to be written to produce a **new** text file, `ClassContact.txt`, containing student names and numbers for all students in a particular class.

For each name contained in the `ClassList` array, the program will:

- search the `StudentContact.txt` file
- copy the matching string into `ClassContact.txt` if the name is found
- write the name together with “*No number” into `ClassContact.txt` if the name is not found.

The program will be implemented as three modules. The description of these is as follows:

| Module | Description |
|---|---|
| `ProcessArray()` | • Check each element of the array:<br>○ Read the student name from the array<br>○ Ignore unused elements<br>○ Call `SearchFile()` with the student name<br>○ If the student name is found, call `AddToFile()` to write the student details to the class file<br>○ If the student name is not found, call `AddToFile()` to write a new string to the class file, formed as follows:<br><Name>“*No number”<br>• Return the number of students who have not provided a telephone number |
| `SearchFile()` | • Search for a given student name at the start of each line in the file `StudentContact.txt`:<br>○ If the search string is found, return the text line from `StudentContact.txt`<br>○ If the search string is not found, return an empty string |
| `AddToFile()` | • Append the given string to a specified file, for example, `AddToFile(StringName, FileName)` |

**(a)** Write **program code** for the module `SearchFile()`.

Visual Basic and Pascal: You should include the declaration statements for variab
Python: You should show a comment statement for each variable used with its data

Programming language ..............................................................................................

Program code

.................................................................................................................................... [8]

**(b)** Write **pseudocode** for the module `ProcessArray()`.

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

.................................................................................................................................... [9]

**(c)** `ProcessArray()` is modified to make it general purpose. It will now be
parameters as follows:

- an array
- a string representing the name of a class contact file

It will still return the number of students who have not provided a contact telephone number.

Write **program code** for the header (declaration) of the modified `ProcessArray()`.

Programming language ..........................................................................................................

Program code

...................................................................................................................................

...................................................................................................................................

...................................................................................................................................

............................................................................................................................. [3]

## Appendix

### Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `ThisString`

Example: `LENGTH("Happy Days")` returns `10`

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns `"ABC"`

`RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns `"FGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.

Example: `NUM_TO_STRING(x)` returns `"87.5"` if `x` has the value `87.5`
Note: This function will also work if `x` is of type `INTEGER`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.

Example: `STRING_TO_NUM(x)` returns `23.45` if `x` has the value `"23.45"`
Note: This function will also work if `x` is of type `CHAR`

### Operators (pseudocode)

| Operator | Description |
|---|---|
| `&` | Concatenates (joins) two strings<br>Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"` |
| `AND` | Performs a logical `AND` on two Boolean values<br>Example: `TRUE AND FALSE` produces `FALSE` |
| `OR` | Performs a logical `OR` on two Boolean values<br>Example: `TRUE OR FALSE` produces `TRUE` |

BLANK PAGE

# QUESTION 15.

**6** Account information for users of a library is held in one of two text files; `UserLis` `UserListNtoZ.txt`

The format of the data held in the two files is identical. Each line of the file is stored as a s contains an account number, name and telephone number separated by the asterisk cha ('`*`') as follows:

```
<Account Number>'*'<Name>'*'<Telephone Number>
```

An example of one line from the file is:

```
    "GB1234*Kevin Mapunga*07789123456"
```

The account number string may be **six** or **nine** characters in length and is **unique for each person**. It is made up of alphabetic and numeric characters only.

An error has occurred and the same account number has been given to different users in the two files. There is **no** duplication of account numbers **within each individual file**.

A program is to be written to search the two files and to identify duplicate entries. The account number of any duplicate found is to be written to an array, `Duplicates`, which is a 1D array of 100 elements of data type `STRING`.

The program is to be implemented as several modules. The outline description of three of these is as follows:

| **Module** | **Outline description** |
|---|---|
| `ClearArray()` | • Initialise the global array `Duplicates`. Set all elements to the empty string. |
| `FindDuplicates()` | • Read each line from the file `UserListAtoM.txt`<br>○ Check whether the account number appears in file `UserListNtoZ.txt` using `SearchFileNtoZ()`<br>○ If the account number does appear then add the account number to the array.<br>• Output an error message and exit the module if there are more duplicates than can be written to the array. |
| `SearchFileNtoZ()` | • Search for a given account number in file `UserListNtoZ.txt`<br>○ If found, return `TRUE`, otherwise return `FALSE` |

**(a)** State **one** reason for storing data in a file rather than in an array.

.......................................................................................................................................

....................................................................................................................................... [1]

**(b)** Write **program code** for the module `SearchFileNtoZ()`.

Visual Basic and Pascal: You should include the declaration statements for variab
Python: You should show a comment statement for each variable used with its data t

Programming language ..........................................................................................

Program code

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..................................................................................................................... [7]

**(c)** Write **pseudocode** for the module `FindDuplicates()`.

The module description is given in the table on page 12.

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

.............................................................................................................................. [8]

**(d)** `ClearArray()` is to be modified to make it general purpose. It will be used to initialise any 1D array of data type `STRING` to any value.

It will now be called with three parameters as follows:

1. The array
2. The number of elements
3. The initialisation string

You should assume that the lower bound is 1.

**(i)** Write **pseudocode** for the modified `ClearArray()` procedure.

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

.............................................................................................................................. [3]

**(ii)** Write **program code** for a statement that calls the modified `ClearArray()` procedure to clear the array `Duplicates` to `"Empty"`.

Programming language ..............................................................................................

Program code

..................................................................................................................................

..................................................................................................................................

[2]

# Appendix

## Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `ThisString`

Example: `LENGTH("Happy Days")` returns `10`

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns `"ABC"`

`RIGHT(ThisString: STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns `"FGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.
Note: This function will also work if `x` is of type `INTEGER`

Example: `NUM_TO_STRING(87.5)` returns `"87.5"`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.
Note: This function will also work if `x` is of type `CHAR`

Example: `STRING_TO_NUM("23.45")` returns `23.45`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns the ASCII value of `ThisChar`

Example: `ASC('A')` returns `65`

`CHR(x : INTEGER) RETURNS CHAR`
returns the character whose ASCII value is `x`

Example: `CHR(87)` returns `'W'`

`UCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the upper case equivalent of `ThisChar`
If `ThisChar` is not a lower case alphabetic character, it is returned unchanged.

Example: `UCASE('a')` returns `'A'`

**Operators (pseudocode)**

| Operator | Description |
|---|---|
| `&` | Concatenates (joins) two strings<br>Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"` |
| `AND` | Performs a logical `AND` on two Boolean values<br>Example: `TRUE AND FALSE` produces `FALSE` |
| `OR` | Performs a logical `OR` on two Boolean values<br>Example: `TRUE OR FALSE` produces `TRUE` |

BLANK PAGE

BLANK PAGE

# QUESTION 16.

**6** A text file, Library.txt, stores information relating to a book collection. The pieces of information about each book on separate lines of the file, as follows:

```
Line n:        <Book Title>
Line n + 1:    <Author Name>
Line n + 2:    <ISBN>
Line n + 3:    <Location>
```

Information is stored as data strings.

Information relating to two books is shown:

| File line | Data |
|---|---|
| 100 | "Learning Python" |
| 101 | "Brian Smith" |
| 102 | "978-14-56543-21-8" |
| 103 | "BD345" |
| 104 | "Surviving in the mountains" |
| 105 | "C T Snow" |
| 106 | "978-35-17635-43-9" |
| 107 | "ZX001" |

**(a)** **(i)** A function, FindBooksBy(), will search Library.txt for all books by a given author.

The function will store the Book Title and Location in the array Result, and will return a count of the number of books found.

Array Result is a global 2D array of type STRING. It has 100 rows and 2 columns.

Write **pseudocode** to declare the array Result.

.............................................................................................................................................

.............................................................................................................................................

............................................................................................................................................. [3]

**(ii)** Function FindBooksBy() will:

- receive the Author Name as a parameter
- search Library.txt for matching entries
- store the Book Title and Location of matching entries in the Result array
- return an integer value giving the number of books by the author that were found.

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

Visual Basic and Pascal: You should include the declaration statements for va
Python: You should show a comment statement for each variable used with its da

Programming language ..............................................................................................

Program code

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..............................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

.......................................................................................................................................... [8]

**(b)** The function `FindBooksBy()` has already been called and has stored values in the array `Result`.

The procedure, `DisplayResults()`, will output the information from the array.

The procedure receives the following two parameters:

- a string containing the author name
- an integer value representing the number of books found

The output should be formatted as in the following example:

```
Books written by: Brian Smith

Title                      Location
Learning Python            BD345
Arrays are not lists       CZ562
Learning Java              CZ589

Number of titles found: 3
```

If no books by the author are found, the following should be output:

```
Search found no books by: Brian Smith
```

Write **pseudocode** for the procedure `DisplayResults()`.

Refer to the **Appendix** on page 16 for the list of built-in functions and operators.

...................................................................................................................................... [7]