basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIOR CERTIFICATE EXAMINATIONS/
NATIONAL SENIOR CERTIFICATE EXAMINATIONS

INFORMATION TECHNOLOGY P1

2021

MARKS: 150

TIME: 3 hours

This question paper consists of 22 pages and 2 data pages.

INSTRUCTIONS AND INFORMATION

1.   This question paper is divided into FOUR sections. Candidates must answer ALL the questions in each of the FOUR sections.

2.   The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.

3.   This question paper is set with programming terms that are specific to the Delphi programming language.

4.   Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.

5.   Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.

6.   Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.

7.   Routines, such as search, sort and selection, must be developed from first principles. You may NOT use the built-in features of Delphi for any of these routines.

8.   All data structures must be declared by you, the programmer, unless the data structures are supplied.

9.   You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.

10.   Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.

11.   If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.

12.   At the end of this examination session you must hand in a disk/CD/DVD/flash disk with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you for this examination session. Ensure that all files can be read.

13.   The files that you need to complete this question paper have been given to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

         NOTE: Candidates must use the file **DataENGJune2021.exe**.

         Do the following:

- Double click on the following password-protected executable file: **DataENGJune2021.exe**.
- Click on the 'Extract' button.
- Enter the following password: **E2021@JUN**

         Once extracted, the following list of files will be available in the folder **DataENGJune2021**:

         FILES PROVIDED:

         Question 1:
Question1_P.dpr
Question1_P.dproj
Question1_P.res
Question1_U.dfm
Question1_U.pas

         Question 2:
ConnectDB_U.pas
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas
ShoppingMallDB - Copy.mdb
ShoppingMallDB.mdb

         Question 3:
GiftVoucher_U.pas
Question3_P.dpr
Question3_P.dproj
Question3_P.res
Question3_U.dfm
Question3_U.pas

         Question 4:
Barbedos.txt
Kensley.txt
Maistry.txt
Question4_P.dpr
Question4_P.dproj
Question4_P.res
Question4_U.dfm
Question4_U.pas

SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

- Open the incomplete program in the **Question 1** folder.
- Enter your examination number as a comment in the first line of the **Question1_U.pas** file.
- Compile and execute the program. The program has no functionality currently.

  Example of the graphical user interface (GUI):
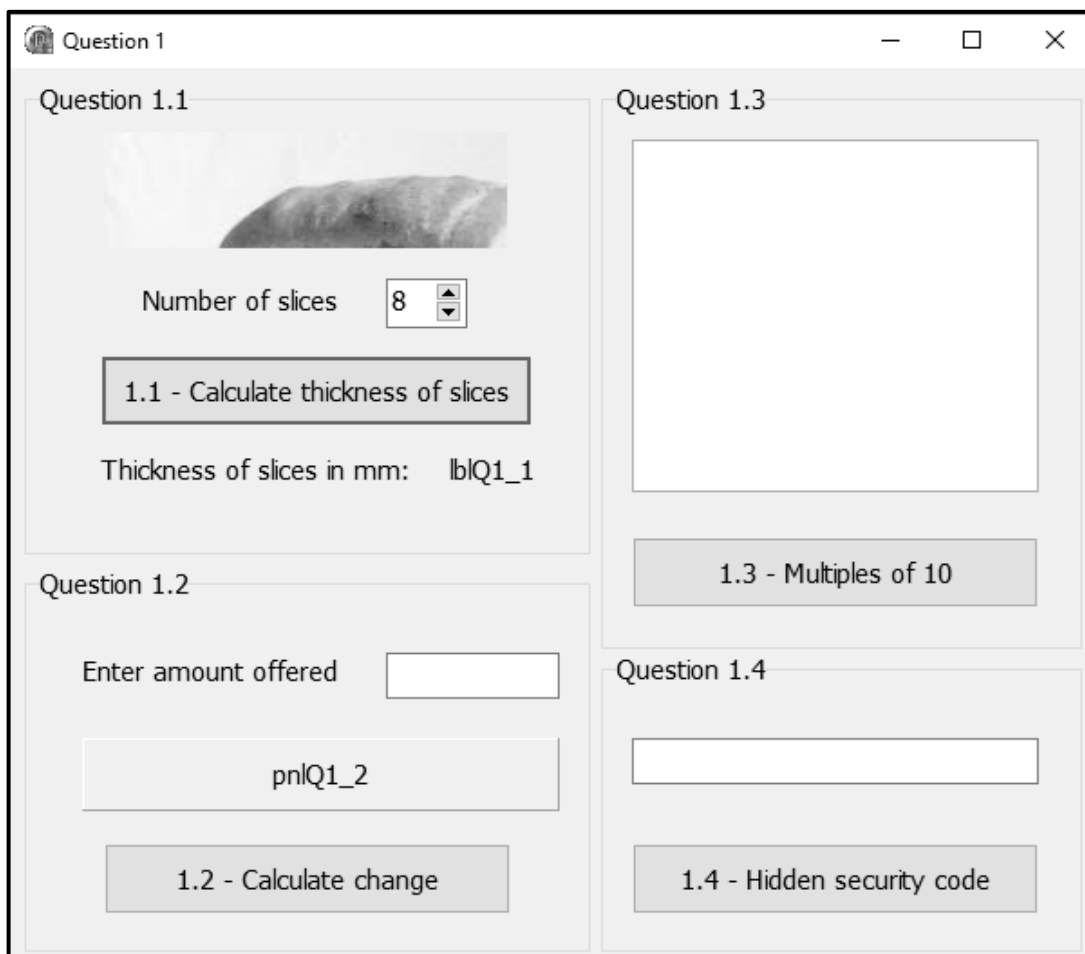
- Complete the code for each section of QUESTION 1, as described in QUESTION 1.1 to QUESTION 1.4 that follow.

1.1   **Button [1.1 - Calculate thickness of slices]**

The length of a loaf of bread is 242 mm. The user is required to select/enter the number of the slices of bread required using the spin edit component. The program must calculate the thickness of the slices.

Write code to do the following:

- Declare appropriate variables of the correct data type for the number of slices and the thickness of the slices. See example output.
- A picture has already been loaded in the **imgQ1_1** image component. Ensure that the full image of the picture is displayed.
- Retrieve the number of slices of bread selected from the **spnQ1_1** spin edit.
- Calculate the thickness of the slices in millimetres (mm).
- Display the thickness of the slices formatted to TWO decimal places in the **lblQ1_1** label.

Example of output if the number of slices selected is 20:
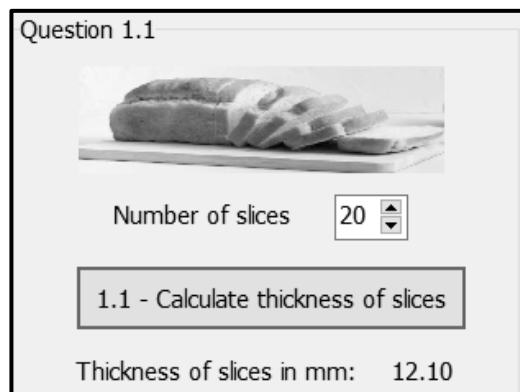
(7)

1.2   **Button [1.2 - Calculate change]**

The cost of a loaf of bread is R12.90. The change needs to be calculated for an amount that is offered to buy a loaf of bread.

Write code to do the following:

- Create a constant, **BREAD_PRICE**, to set the price of a loaf of bread to 12.90. The constant must be used when referring to the price of a loaf of bread in the code.
- Declare two variables of real/double data type to store the amount offered by the customer and the change.
- Retrieve the amount offered from the **edtQ1_2** edit box.
- Display the change in currency format in the panel **pnlQ1_2** if the amount offered is more than or equal to the BREAD_PRICE.
- If the amount offered is less than BREAD_PRICE, display a suitable message in the panel **pnlQ1_2**.

Example of output if an amount of R15.00 was offered:

```
Question 1.2

Enter amount offered    [15.00    ]

        Change: R2.10

     1.2 - Calculate change
```

Example of output if an amount of R10.00 was offered:

```
Question 1.2

Enter amount offered    [10.00    ]

    Insufficient amount offered

     1.2 - Calculate change
```

Example of output if an amount of R12.90 was offered:

```
Question 1.2

Enter amount offered    [12.90    ]

        Change: R0.00

     1.2 - Calculate change
```
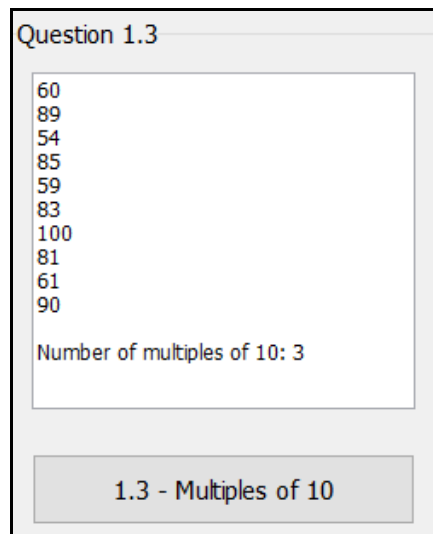
(11)

1.3   **Button [1.3 - Multiples of 10]**

Write the code to do the following:

- Use a loop structure to generate 10 random values in the range 50 to 100 (inclusive). Display the values in the **redQ1_3** output area.
- Count and display the number of random values that are multiples of 10.

Example of output:

NOTE: The values in the example output may differ from your output as the values are randomly generated.

(10)

1.4   **Button [1.4 - Hidden security code]**

Two variables, **sParagraph** and **sSecurityCode**, have been declared in the code. A paragraph of text, which contains a security code, has been assigned to the variable sParagraph. Rules need to be followed to extract specific characters from the text in the sParagraph variable to compile a security code which must be saved in the provided variable sSecurityCode.

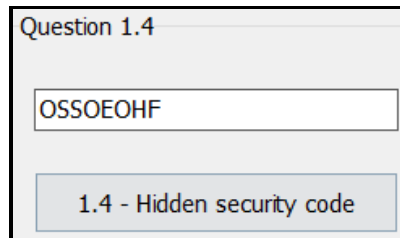The following rules must be applied to compile the security code:

- The character before every letter 't' (lower case) in sParagraph must be extracted and combined to form a security code.
- If the character before a letter 't' is a space, it must be ignored and must NOT form part of the security code.
- The search for characters for the security code must start at the beginning of the paragraph.
- The maximum length of the security code is 8 characters.
- The security code must be displayed in upper case in the edit box **edtQ1_4**.

Write code to compile the security code using a loop structure and by applying the specified rules.

NOTE: Your code must output the correct security code for any text assigned to the sParagraph variable.

Example of output if the following text is allocated to sParagraph:

sParagraph = 'I am not lazy, I am just very relaxed. He who laughs last did not get the joke. When nothing is going right, go left. I love school when it is vacation. I put the "Pro" in procrastinate.'
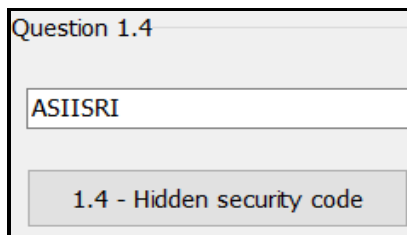
Question 1.4

OSSOEOHF

1.4 - Hidden security code

Example of output if the following text is allocated to sParagraph:

sParagraph = 'My Grade 12 year is always going to be my greatest as it provides me with the most opportunities.'

Question 1.4

ASIISRI

1.4 - Hidden security code

(12)

- Ensure that your examination number has been entered as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

TOTAL SECTION A: 40

SECTION B

QUESTION 2: SQL AND DATABASE

> The database **ShoppingMallDB** contains information on different types of shops in a shopping mall and the details of the managers of the shops. The database contains two tables, namely **tblShops** and **tblManagers**. Each manager is in charge of a specific type of shop, e.g. Emily Farez is in charge of all clothing shops.

The data pages attached at the end of the question paper provide information on the design of the database and the content of the tables.

Do the following:

- Open the incomplete program in the **Question 2** folder.
- Enter your examination number as a comment in the first line of the **Question2_U.pas** file.
- Compile and execute the program. The program has no functionality currently. The contents of the tables are displayed as shown below on the selection of tab sheet **Question 2_2 Delphi code**.

| Question 2.1 - SQL | Question 2.2 - Delphi code |

tblManagers

ManagerNumber	ManagerName	ManagerSurname	ContactNumber
1	Peter	Ndlovu	0825608814
2	Sandra	Nell	0834567892
3	Emily	Farez	0831564789

tblShops

ShopNumber	ShopName	ShopSize	TurnOver	ManagerNumber
101	Little Kitchen Grocery Store	652	R966 826.00	1
102	Foodies Groceries	642	R688 217.00	1
103	Lorem Company	538	R814 141.00	1
104	I&J Store	590	R559 383.00	1

- Follow the instructions below to complete the code for each section, as described in QUESTION 2.1 and QUESTION 2.2.
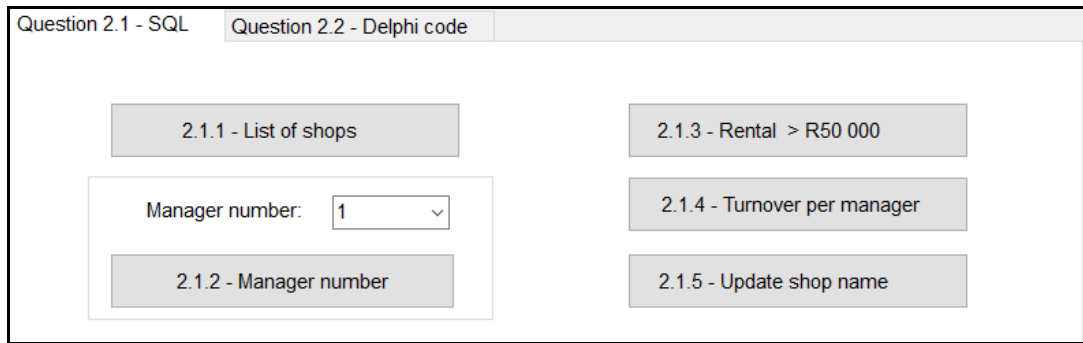
NOTE:

- The 'Restore database' button is provided to restore the data contained in the database to the original content.
- Code is provided to link the GUI components to the database. Do NOT change any of the code provided.
- TWO variables are declared as public variables, as described in the table on the next page.

Variable	Data type	Description
`tblShops`	`TADOTable`	Refers to the table tblShops
`tblManagers`	`TADOTable`	Refers to the table tblManagers

2.1 Tab sheet [Question 2.1 - SQL]

Example of graphical user interface (GUI) for QUESTION 2.1:

NOTE:

- Use only SQL code in this section.
- Code is provided to execute the SQL statements and to display the results of the queries. The SQL statements assigned to the variables **sSQL1**, **sSQL2**, **sSQL3**, **sSQL4** and **sSQL5** are incomplete.

Complete the SQL statements to perform the tasks described in QUESTION 2.1.1 to QUESTION 2.1.5 that follow.

2.1.1 Button [2.1.1 - List of shops]

Display ALL fields of the shops in the **tblShops** table, sorted from the largest to the smallest shop according to the **ShopSize** field.

Example of output of the first four records:

ShopNumber	ShopName	ShopSize	TurnOver	ManagerNumber
402	Mauris Chicken	991	R912 199.00	4
105	Fetch and Go Groceries	926	R240 340.00	1
303	Gorgeous Designs	892	R988 452.00	3
407	FreeRange Steakhouse	803	R954 777.00	4

(3)

2.1.2 Button [2.1.2 - Manager number]

The user must select a manager number from the combo box **cmbQ2_1_2.** Code has been provided to assign the manager number selected by the user to a variable **sManagerNum**. Display the **ShopName** of all the shops managed by the manager associated with the selected manager number.

Example of output if manager number 1 is selected:

ShopName
Little Kitchen Grocery Store
Foodies Groceries
I&J Store
Fetch and Go Groceries
Massa Cash & Carry
Lorem Company

(3)

2.1.3   Button [2.1.3 - Rental > R50 000]

The rental cost of space in the mall is R65.00 per square metre. Use the **ShopSize** field, which is in square metres, to calculate the rental costs for each shop. Use the new field named **Rental** to keep the rent amounts.

Display the shop name, shop size and calculated rental amount of all shops with rental amounts that exceed the value of R50 000. The rental amount must be formatted as currency.

Example of output:

ShopName	ShopSize	Rental
Fetch and Go Groceries	926	R60 190.00
E&E Electronics	793	R51 545.00
Gorgeous Designs	892	R57 980.00
Velit Men's Wear	798	R51 870.00
Mauris Chicken	991	R64 415.00
FreeRange Steakhouse	803	R52 195.00

(5)

2.1.4   Button [2.1.4 - Turnover per manager]

The total turnover is calculated by adding the values in the **Turnover** field. The total turnover must be calculated per manager.

Display the **ManagerName**, **ManagerSurname** and the calculated field called **TotalTurnover** for EACH manager.

Example of output:

ManagerName	ManagerSurname	TotalTurnover
Emily	Farez	R3 078 782.00
Mohamed	Khan	R2 882 898.00
Peter	Ndlovu	R4 171 066.00
Sandra	Nell	R1 928 425.00
Zondile	Ngobeni	R2 332 710.00

(6)

2.1.5   **Button [2.1.5 - Update shop name]**

The clothing shop, Jeans 4U, has moved to another shopping mall. A new clothing store with the name TeenDream opened in its place. Modify the relevant record by changing the shop name Jeans 4U to TeenDream.
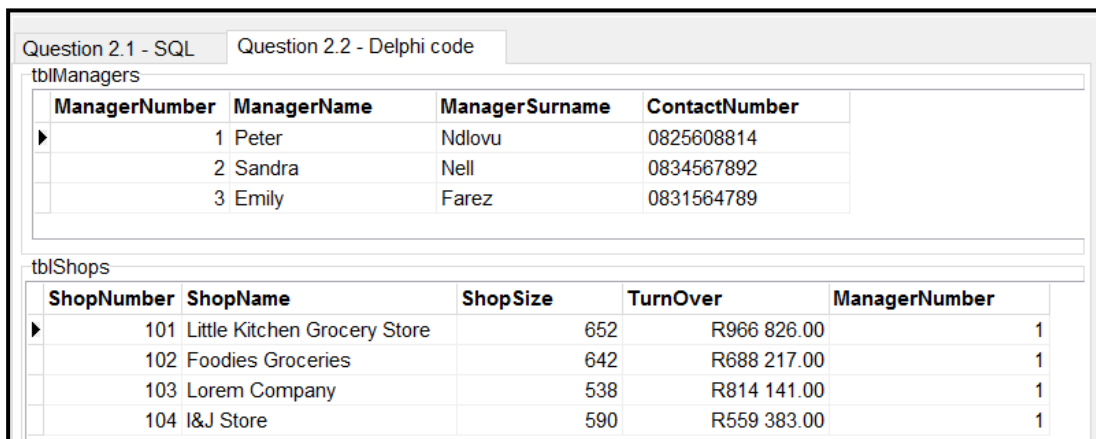
Example of output:

```
Question2_p                                    ×

Database updated

                                      [  OK  ]
```

(3)

2.2   **Tab sheet [Question 2.2 - Delphi code]**

Example of graphical user interface (GUI) for QUESTION 2.2:

tblManagers

ManagerNumber	ManagerName	ManagerSurname	ContactNumber
1	Peter	Ndlovu	0825608814
2	Sandra	Nell	0834567892
3	Emily	Farez	0831564789

tblShops

| ShopNumber	ShopName	ShopSize	TurnOver	ManagerNumber
101	Little Kitchen Grocery Store	652	R966 826.00	1
102	Foodies Groceries	642	R688 217.00	1
103	Lorem Company	538	R814 141.00	1
104	I&J Store	590	R559 383.00	1

NOTE:

- Use ONLY Delphi programming code to answer QUESTION 2.2.1 and QUESTION 2.2.2.
- NO marks will be awarded for SQL statements in QUESTION 2.2.

2.2.1   **Button [2.2.1 - Number of small shops]**

A shop is classified as a small shop if the size of the shop is less than 300 square metres.

Write code to count and display the number of small shops in the rich edit component **redQ2_2_1**.
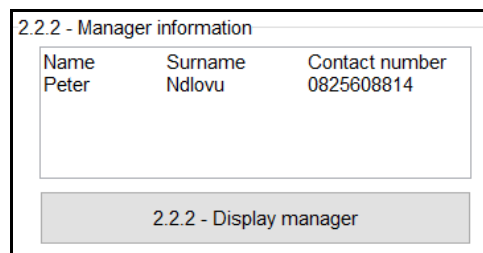
Example of output:
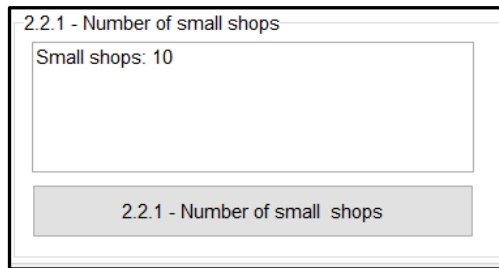
(7)

2.2.2 **Button [2.2.2 - Display manager]**

Information on the manager of a shop must be displayed on request. Code has been provided to enter the name of a shop using an input box.

Write code to do the following:

- Search the relevant table to find the name of the shop that was entered.
- If the name of the shop is found, display the **name**, **surname** and **contact number** of the manager of the shop in the rich edit component **redQ2_2_2**.
- If the name of the shop is NOT found, use a message dialogue box to display the message 'The shop is not in this mall.'

Example of output if the name of the shop 'Little Kitchen Grocery Store' was entered and found:

Example of output if the name of the shop that was entered was not found:
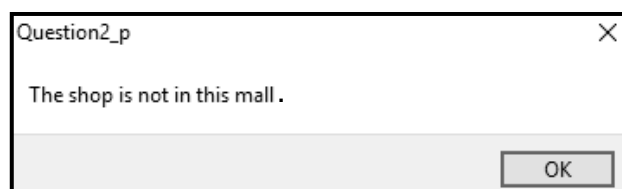
(13)

- Ensure that your examination number has been entered as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

TOTAL SECTION B: 40

SECTION C

QUESTION 3: OBJECT-ORIENTATED PROGRAMMING

The MidCity Shopping Centre sells gift vouchers that can be used at any shop in the shopping centre. The minimum amount available per gift voucher card is R50 and the maximum amount available is R300.

Do the following:

- Open the incomplete program in the **Question 3** folder.
- Open the incomplete object class **GiftVoucher_U.pas**.
- Enter your examination number as a comment in the first line of both the **Question3_U.pas** file and the **GiftVoucher_U.pas** object class file.
- Compile and execute the program. The program has limited functionality currently.

Example of the graphical user interface (GUI):
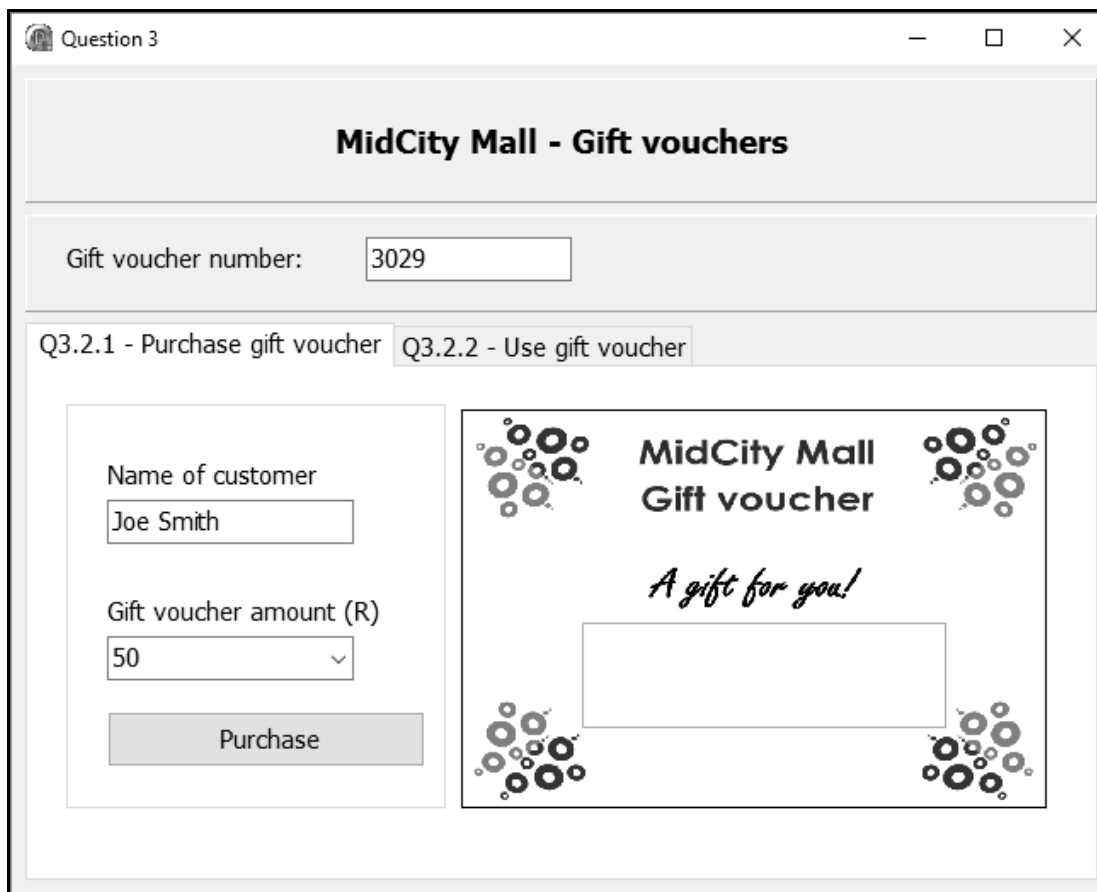
Complete the code as specified in QUESTION 3.1 and QUESTION 3.2.

NOTE: For this question, you are NOT allowed to include any additional attributes or user-defined methods not stated in the question.

3.1   The provided incomplete object class (**TGiftVoucher**) contains code for the declaration of three attributes that describe a **GiftVoucher** object.

The attributes for a **GiftVoucher** object have been declared as follows:

- fVoucherNumber – a four-digit integer value that represents the number of the gift voucher
- fName – the name of the customer who is buying and using the voucher
- fBalance – a real value that contains the amount still available on the gift voucher

Code has been provided for an incomplete **toString** method.

Complete the code in the object class as described in QUESTION 3.1.1 to QUESTION 3.1.5.

3.1.1   Write code for a **constructor** method that will receive the gift voucher number, the name of the customer and the balance (amount available on gift voucher) as parameter values. Assign these values to the respective attributes.   (4)

3.1.2   Write code for two accessor methods called **getVoucherNumber** and **getBalance** for the **fVoucherNumber** and **fBalance** attributes respectively.   (4)

3.1.3   Write code for a method called **isSufficient** to receive a real value for a purchase amount and return a Boolean value TRUE if the balance on the gift voucher is sufficient to pay the purchase amount, or FALSE if not.   (5)

3.1.4   Write code for a method called **updateBalance** that will receive a purchase amount as a parameter and subtract the purchase amount from the balance.   (3)

3.1.5   Write code to complete the **toString** method to return a string in the following format:

```
Voucher number: <fVoucherNumber>
Customer name: <fName>
Available balance: <fBalance>
```

Example:

```
Voucher number: 3029
Customer name: Joe Smith
Available balance: R250.00
```
                                                                                                      (4)

3.2   An incomplete program has been supplied in the **Question 3** folder. The program contains code for the object class to be accessible and declares an object variable called **objGiftVoucher**.

Write code to perform the tasks described in QUESTION 3.2.1 and QUESTION 3.2.2 to purchase and use a gift voucher. The GUI contains two separate tab sheets for QUESTION 3.2.1 and QUESTION 3.2.2 respectively.

3.2.1   Tab sheet Q3.2.1 - Purchase gift voucher

Button [Purchase]

The user must enter a voucher number in the edit box **edtQ3_VoucherNum**, enter the name of the customer in the edit box **edtQ3_2_1** and select an amount to be loaded onto the gift voucher from the combo box **cmbQ3_2_1**.

Write code to do the following:

- Extract the gift voucher number, the name of the customer and the amount from the relevant components.
- Use the extracted values to instantiate a new **objGiftVoucher** object.
- Display the details of the gift voucher in the output area **redQ3_2_1** using the **toString** method.

Example of input and output if a gift voucher with the value of R250 was selected for Joe Smith:

(6)

3.2.2   **Tab sheet Q3.2.2 - Use gift voucher**

(a)   **Button [Display balance]**

To be able to check the balance of a gift voucher, the gift voucher number must be entered in the **edtQ3_VoucherNum** edit box.

Write code to do the following:

- The gift voucher number to be used must be retrieved from the edit box **edtQ3_VoucherNum**. Test if the gift voucher number entered is the same as the gift voucher number of the **objGiftVoucher** object by comparing it to the relevant object attribute value using the relevant object method.
- If the gift voucher numbers are the same:

   o   Use the **getBalance** method to display the gift voucher balance in currency format in panel **pnlQ3_2_2**.
   o   Enable the **btnQ3_2_2_b** button.

Example of input and output:

(5)

(b)   **Button [Use gift voucher]**

A customer can use the gift voucher when making a purchase at the mall. When using the voucher, the amount for the purchase must be entered in the edit box provided.

Write code to do the following:

- Extract the purchase amount from the edit box **edtQ3_2_2**.
- Use the **isSufficient** method to determine whether the gift voucher amount can cover the purchase amount.

o   If the amount on the gift voucher is sufficient, display the message, 'Gift voucher used successfully' in the label **lblQ3_2_2** and call the **updateBalance** method with the correct argument to update the balance.
o   If the amount on the gift voucher is not sufficient to cover the purchase amount:
    ▪   Display the message 'Amount owed by you to complete the purchase –', followed by the outstanding amount formatted as currency in the label **lblQ3_2_2**.
    ▪   Call the **updateBalance** method with the correct argument to change the balance attribute to zero.

•   Display the new balance in the panel **pnlQ3_2_2**.

Example of output if the purchase amount is **less than the balance** on the gift voucher:

Before the transaction:

BALANCE ON GIFT VOUCHER
R250.00

After the gift voucher has been used:

BALANCE ON GIFT VOUCHER

| **R117.12** | | Display balance |

Transaction

Enter purchase amount   132.88

Gift voucher used successfully

Example of output if the purchase amount is **equal to the balance** on the gift voucher:

Before the transaction:

BALANCE ON GIFT VOUCHER
R250.00

After the gift voucher has been used:

BALANCE ON GIFT VOUCHER

| **R0.00** | | Display balance |

Transaction

Enter purchase amount   250

Gift voucher used successfully

Example of output if the purchase amount is **more than the balance** on the gift voucher:

Before the transaction:

BALANCE ON GIFT VOUCHER
R250.00

After the gift voucher has been used:

BALANCE ON GIFT VOUCHER	
R0.00	Display balance

Transaction

Enter purchase amount   357.50

Amount owed by you to complete the purchase - R107.50

(9)

- Ensure that your examination number has been entered as a comment in the first line of the object class and the form class.
- Save your program.
- Print the code in the object class and the form class if required.


TOTAL SECTION C: 40

SECTION D

QUESTION 4: PROBLEM-SOLVING PROGRAMMING

Three new malls have been built recently. Each mall has a ground floor that has demo shops to showcase the types of shops on the different levels at the mall. For example, if the ground floor has a clothing store, a restaurant and a furniture store, then the other levels in the mall can only have the types of shops as showcased on the ground floor.

Do the following:

- Open the incomplete program in the **Question 4** folder.
- Enter your examination number as a comment in the first line of the **Question4_U.pas** file.
- Compile and execute the program. The program has no functionality currently.

Example of the graphical user interface (GUI):
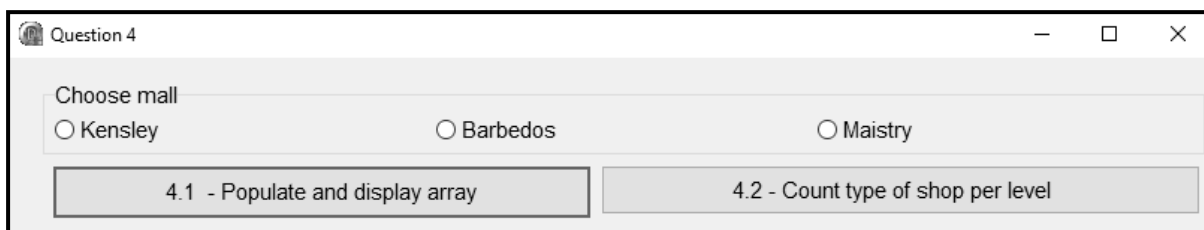
The following have been provided in the program:

- A one-dimensional array, **arrShopTypes**, consisting of the descriptions of 8 types of shops that can be at the mall.

  arrShopTypes: array [1..8] of String = ('Electronics', 'Jewellery', 'Furniture', 'Clothing', 'Restaurant', 'Toys', 'Laundry', 'Gifts');

- Three text files, one for each of the three different malls (Kensley, Barbedos, Maistry). Each line of text in a text file represents a level in the mall. The characters that a line of text consists of indicate the types of shops on a specific level in the mall. For example, the text file named 'Kensley' contains six lines of text which means there are six levels:

  LEJCF
  JCFERECFLE
  EJCFFECCFFEERRR
  JECFFRRJECFFRRE
  LEJCFLLLL
  FFLEJJJJCCCF

Each character in a line of text represents a type of shop (the first letter of the given list of shop types in array **arrShopTypes**) that can be found on that specific level in the mall. For example, the letter E refers to Electronics, J refers to Jewellery and so on. The characters used are E, J, F, C, R, T, L and G.

The second line of text in the text file for the Kensley Mall contains the following line of text, 'JCFERECFLE', which indicates that there are 10 shops on level 2 of the mall – one jewellery shop (J), two clothing shops (C), two furniture shops (F), three electronics shops (E), one restaurant (R) and one laundry (L).

Write code to perform the tasks described in QUESTION 4.1 to QUESTION 4.2.

4.1   **Button [4.1 - Populate and display array]**

The user must select a mall from the radio group **rgpQ4**.

Write code to do the following:

- Use the text of the radio button selected to compile the file name which indicates which text file to read.
- Read the contents of the text file into an array. The maximum number of levels in a mall is ten.
- Display the lines of text representing the type of shops in the mall per level and the number of shops per level, as shown in the screenshot below.
- The display of the shops must be right-justified. You can assume that the maximum length of a line of text with the types of shops, including the spaces, is 25.

Example of output if the 'Kensley' mall was selected:

```
Level:             Shops:     Number of shops per level:
==========================================================
1.                  LEJCF     5
2.             JCFERECFLE     10
3.        EJCFFECCFFEERRR     15
4.        JECFFRRJECFFRRE     15
5.              LEJCFLLLL     9
6.           FFLEJJJJCCCF     12
```
(15)

4.2   **Button [4.2 – Count shop type per level]**

An enquiry about the number of shops of a specific type per level in the mall must be processed and displayed. The user must enter a character representing the type of shop using an input box.

Show an error message if the input character is not in the range of valid characters, namely E, J, F, C, R, T, L and G.

If a valid character is entered, the number of the specified type of shop in the selected mall must be determined per level and neatly displayed, as shown in the examples below. The heading of the output must display the description of the type of shop in the **arrShopTypes** array.

Example of output if the Kensley Mall was selected and F was selected as the shop type:

```
Level:          Shops:     Number of shops per level:
===========================================================
1.              LEJCF      5
2.          JCFERECFLE     10
3.      EJCFFECCFFEERRR    15
4.      JECFFRRJECFFRRE    15
5.           LEJCFLLLL     9
6.         FFLEJJJJCCCF    12

Type of shop: Furniture
Number per level
1.  1
2.  2
3.  4
4.  4
5.  1
6.  3
```

Example of output if the Barbedos Mall was entered and C was selected as the shop type:

```
Level:          Shops:     Number of shops per level:
===========================================================
1.          EJCFFECCEECF    12
2.           JCFERECFTT     10
3.      JECFFRRJFFRRECEJ    16
4.              RRJCE      5
5.               JEC       3
6.      EJFCRTEJFCRTTTTREE  18
7.  JCFERECFTTJCFERECFTT    20
8.       RRJCERRJCERRJCE    15

Type of shop: Clothing
Number per level
1.  4
2.  2
3.  2
4.  1
5.  1
6.  2
7.  4
8.  3
```

(15)

- Ensure that your examination number has been entered as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

TOTAL SECTION D: **30**
GRAND TOTAL: **150**

INFORMATION TECHNOLOGY P1

DATABASE INFORMATION QUESTION 2:

The design of the database tables is as follows:

Table: **tblManagers**

The table contains the records of the managers who manage each court in the shopping mall.

Field name	Data type	Description
ManagerNumber (PK)	Byte	A unique number assigned to each manager
ManagerName	Text (30)	The name of the manager
ManagerSurname	Text (30)	The surname of the manager
ContactNumber	Text (10)	The contact number of the manager

Example of the first five records of the **tblManagers** table:

ManagerNumber	ManagerName	ManagerSurname	ContactNumber
1	Peter	Ndlovu	0825608814
2	Sandra	Nell	0834567892
3	Emily	Farez	0831564789
4	Mohamed	Khan	0715658360
5	Zondile	Ngobeni	0716358922

Table: **tblShops**

The table contains the information of the shops located in the shopping mall.

Field name	Data type	Description
ShopNumber (PK)	Number	A unique number assigned to each shop
ShopName	Text (50)	The name of the shop
ShopSize	Number	The size of the shop
TurnOver	Currency	The turnover of the shop
ManagerNumber (FK)	Byte	A number that identifies the manager of a shop

Example of the first ten records in the **tblShops** table:

ShopNumber	ShopName	ShopSize	TurnOver	ManagerNumber
101	Little Kitchen Grocery Store	652	R966 826.00	1
102	Foodies Groceries	642	R688 217.00	1
103	Lorem Company	538	R814 141.00	1
104	I&J Store	590	R559 383.00	1
105	Fetch and Go Groceries	926	R240 340.00	1
106	Massa Cash & Carry	492	R902 159.00	1
201	CJ Cell Repairs	94	R706 979.00	2
202	E&E Electronics	793	R315 584.00	2
203	Molestie PC	293	R225 334.00	2
204	Software Designs	760	R312 062.00	2

NOTE:

- Connection code has been provided.
- The database is password-protected, therefore you will not be able to access the database directly.

The following one-to-many relationship with referential integrity exists between the two tables in the database:

tblManagers

- ManagerNumber
- ManagerName
- ManagerSurname
- ContactNumber

1 ———— ∞

tblShops

- ShopNumber
- ShopName
- ShopSize
- TurnOver
- ManagerNumber