Province of the
EASTERN CAPE
EDUCATION

Iphondo leMpuma Kapa: Isebe leMfundo
Provinsie van die Oos Kaap: Department van Onderwys
Porafensie Ya Kapa Botjhabela: Lefapha la Thuto

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2024

INFORMATION TECHNOLOGY P1

MARKS: **150**

TIME: **3 hours**

This question paper consists of 25 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into FOUR questions. Candidates must answer ALL the questions.

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 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 a programming language for any of these routines.

8. All data structures must be defined 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 name 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. Make sure that all files can be read.

13. Save your work regularly.

14. The files that you need to complete this question paper have been provided 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.

Do the following:

- Double click on the password-protected executable file:
- **DataENGAug2024.exe**
- Click on the 'Extract' button.
- Enter the following password: #**Crypto@Aug2024**

Once extracted, the following list of files will be available in the folder **DataENGAug2024**:

Question 1:

Question1_P.dpr
Question1_P.dproj
Question1_P_Icon.ico
Question1_U.dfm
Question1_U.pas

Question 2:

CryptoExchange.mdb
CryptoExchangeBackUp.mdb
dbConnection_u.pas
Question2_P.dpr
Question2_P.dproj
Question2_P_Icon.ico
Question2_U.dfm
Question2_U.pas

Question 3:

Crypto_U.pas
Question3_P.dpr
Question3_P.dproj
Question3_P_Icon.ico
Question3_U.dfm
Question3_U.pas

Question 4:

Crypto.txt
Question4_P.dpr
Question4_P.dproj
Question4_P_Icon.ico
Question4_U.dfm
Question4_U.pas

QUESTION 1: GENERAL PROGRAMMING SKILLS

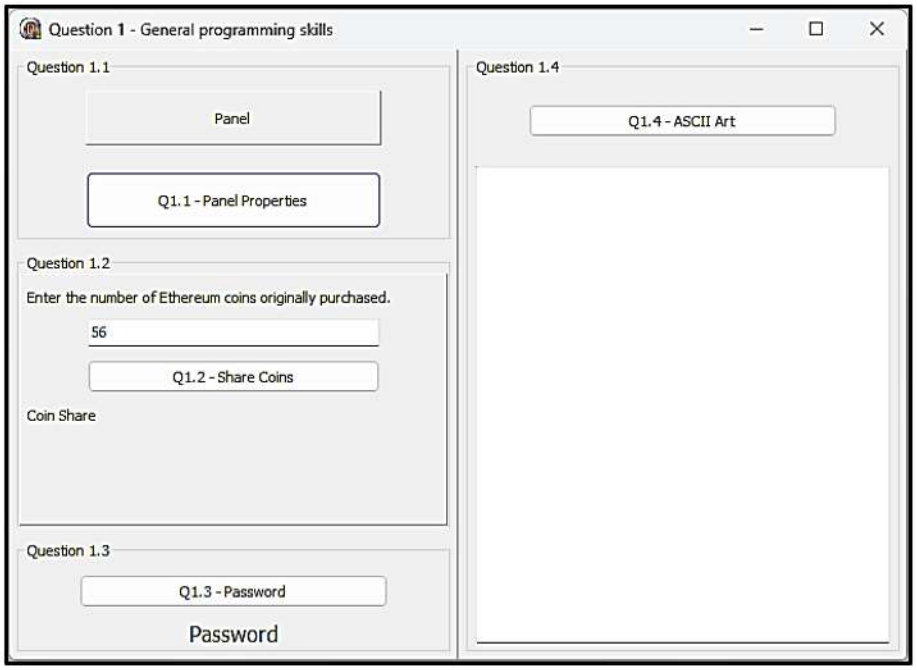
SCENARIO

Cryptocurrencies have once again surged in popularity. Given the rapid advancement in existing and emerging cryptocurrency technologies, grasping the fundamentals, security protocols, and practical applications is imperative. Your task is to devise programming solutions for the following questions pertaining to cryptocurrencies.

Do the following:

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

Example of the graphical user interface (GUI):



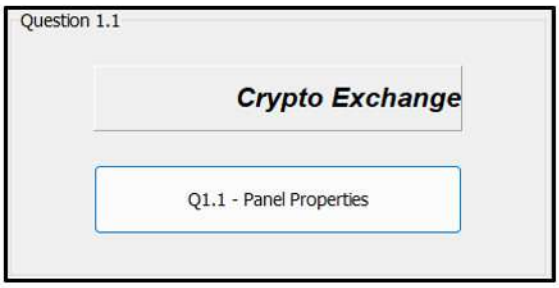
Complete the code for each section of QUESTION 1, as described in QUESTION 1.1 to QUESTION 1.4 that follow.

1.1 Button [1.1 – Panel Properties]

Code the button **btn1_1** to change the properties of panel **pnlOutput** as follows:

1. Set the font type to Arial.
2. Set the font size to 16.
3. Set the font to Bold and Italic.
4. Set the text to display on the right hand side of the panel.
5. Change the caption to 'Crypto Exchange'.

Example of output:



(6)

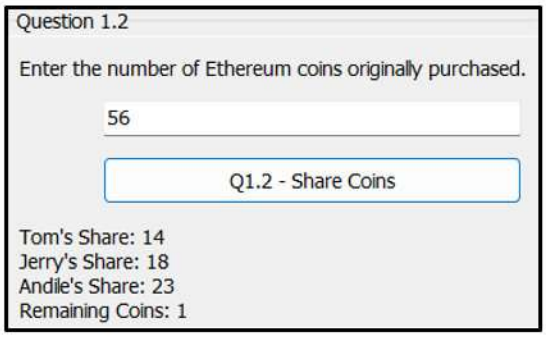
1.2 **Button [1.2 – Share Coins]**

Three friends Tom, Jerry, and Andile, purchased Ethereum coins when the price was still low. They decide to 'cash out' now. You need to share the coins in the ratio 3 : 4 : 5 respectively. The number of coins bought is provided by the user in the edit box **edtInput**. If there are remaining coins, they will not be shared amongst the friends but rather left in the exchange for growth.

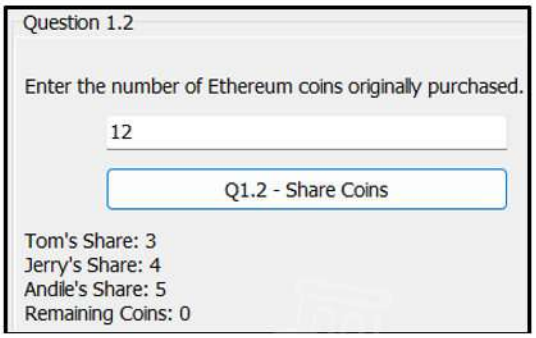
*****Note that each coin will be treated as a full coin and no decimal values should be produced or displayed.******

Code the button **btn1_2** to calculate how many coins each friend will receive, and how many coins will remain after sharing takes place. Output in the label **lbl1_2** as per example screenshots. Ensure the correct use of apostrophes and enter spaces.

Output if 56 coins were purchased.



Output if 12 coins were purchased.



(10)

1.3 **Button [Q1.3 – Password]**

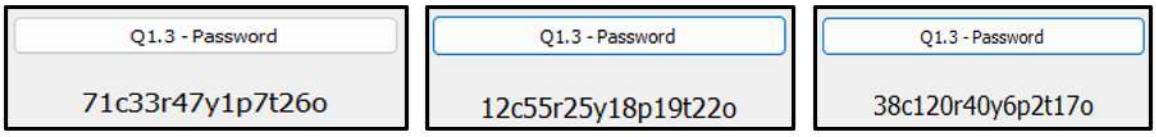
Crypto accounts require strong passwords. You are thus required to write an algorithm that will generate a strong password.

Code the button **btn1_3** to generate a password based on the following rules.

The password must contain the letters “CRYPTO”.

1. Loop through each letter of the word “CRYPTO”.
2. For each letter, randomly generate characters from ‘A’ to ‘Z’ until you find a character that matches the corresponding letter in “CRYPTO”.
3. Count the number of randomisations required to find the matching character.
4. Append the count of randomisations followed by the matched character (must be lowercase) to the password.
5. Repeat steps 2–4 for each letter in “CRYPTO”.
6. Display the generated password in the label **lbl1_3**. (12)

Example of output *(**NOTE:** each output will differ due to the random counts of each character).*



1.4 **Button [Q1.4 – ASCII Art]**

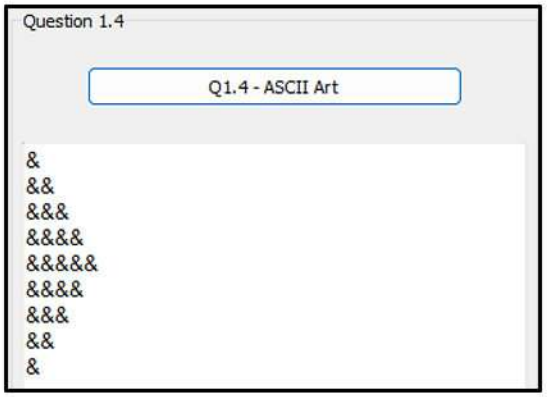
Some of the MEME coins would like their branding to involve ASCII Art (art made from characters).

Code the button **btn1_4** to generate a pattern based on the following rules:

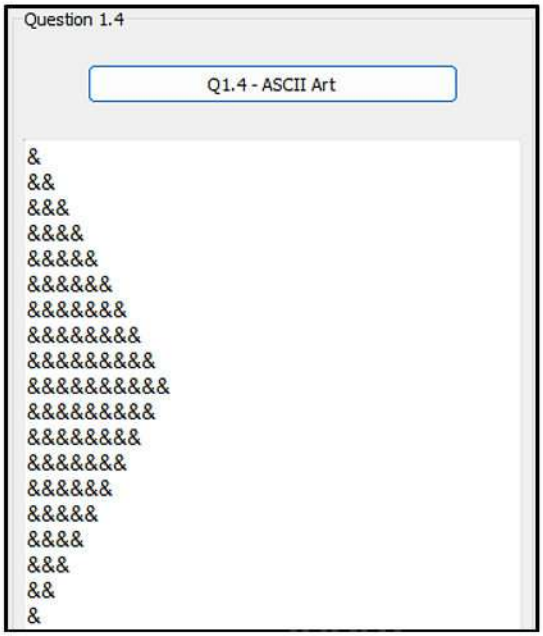
1. Prompt the user to input the length of the longest line for the pattern.
2. Construct the pattern using the character '&'.
3. Begin with a single '&' character on the first line.
4. For each subsequent line, add one '&' character to the end of the previous line.
5. After reaching the maximum number of characters, begin removing '&' characters from the beginning of each line until only one '&' character remains on the last line.
6. Display the generated pattern in the rich edit **redOutput**.

Example output:

Output for maximum length of 5.



Output for maximum length of 10.



(7)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[35]

QUESTION 2: SQL AND DATABASE PROGRAMMING

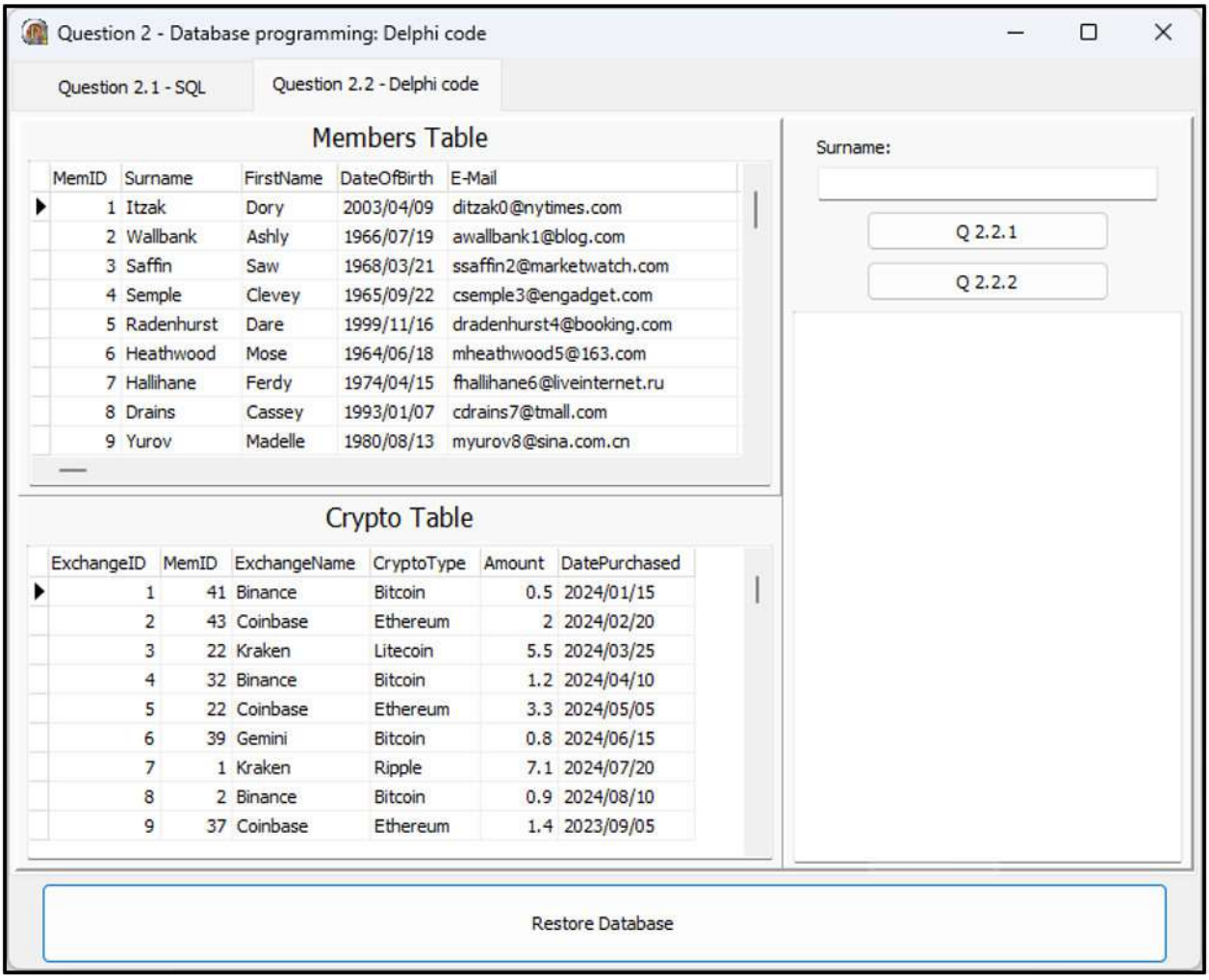
An application is required that will use the **CryptoExchange.mdb** database to manage the data and queries of cryptocurrency traders.

The database contains two tables called **tblMembers** and **tblCrypto**.

The data pages attached at the end of the question paper provide information on the design of the **CryptoExchange.mdb** database and its contents.

Do the following:

- Open the incomplete program in the **Question 2** folder.
- Enter your full name as a comment in the first line of the **Question2_U.pas** unit file.
- Compile and execute the program. The program has limited functionality.
- The contents of the tables are displayed as shown below on the selection of tab sheet **Question 2.2**



- Follow the instructions below to complete the code for each section as described in QUESTION 2.1 and QUESTION 2.2.
- Use SQL statements to answer QUESTION 2.1 and Delphi code to answer QUESTION 2.2.

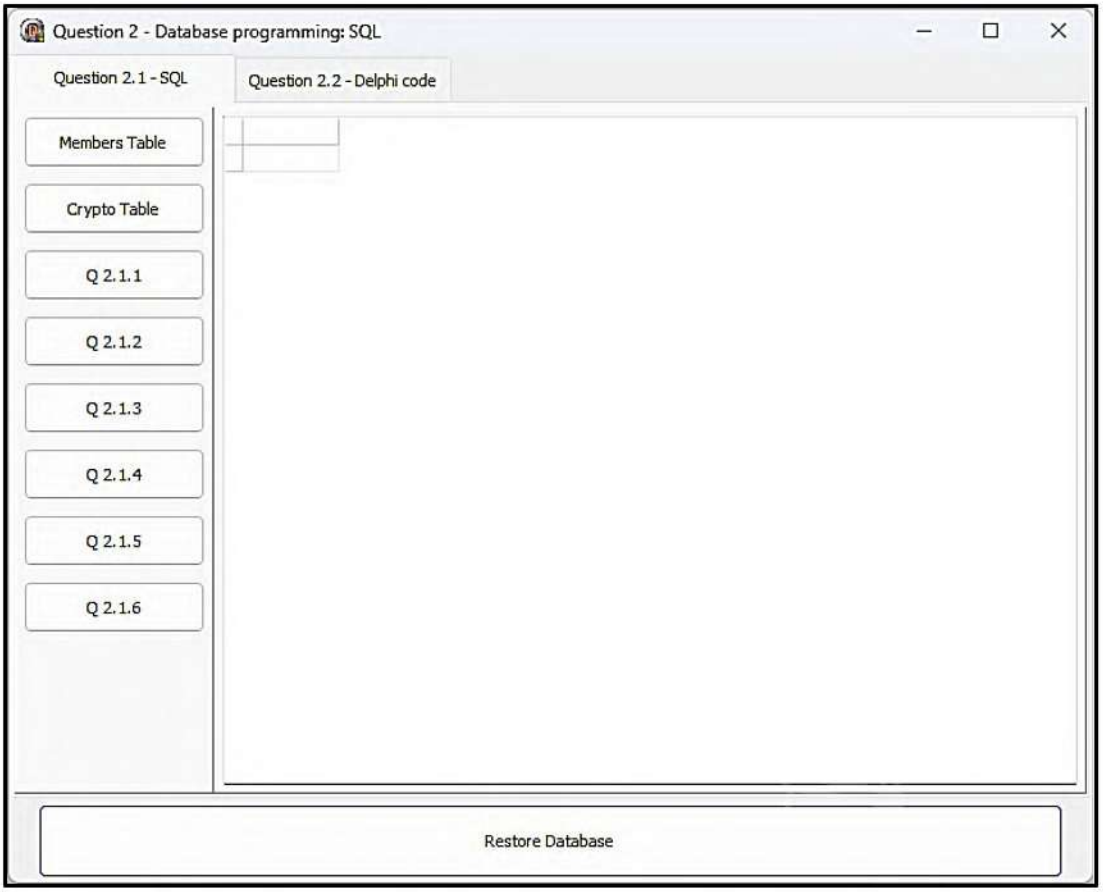
NOTE:

- The 'Restore database' button is provided to restore the data contained in the database to the original content.
- The content of the database is password-protected, i.e. you will NOT be able to gain access to the content of the database using Microsoft Access.
- Code is provided to link the GUI components to the database. Do NOT change any of the code provided.
- THREE variables are declared as public variables, as described in the table below:

Variable	Data type	Description
tblCrypto	TADOTable	Refers to the data stored in the table tblCrypto
tblMembers	TADOTable	Refers to the data stored in the table tblMembers
qryInfo	TADOQuery	Query component that will query the two tables tblCrypto and tblMembers

2.1 Tab Sheet [Question 2.1]

Example of the graphical user interface (GUI) for QUESTION 2.1.



NOTE:

- Use ONLY SQL code to answer QUESTION 2.1.1 to QUESTION 2.1.6.
- Code to execute the SQL statements and display the results of the queries is provided. The SQL statements that will be assigned to the variables **sSQL1, sSQL2, sSQL3, sSQL4, sSQL5,** and **sSQL6** are incomplete.

Complete the SQL statements to perform the tasks described in QUESTION 2.1.1 to QUESTION 2.1.6 below.

2.1.1 Button [Q2.1.1]

Write SQL code to display all the fields of all the members, sorted by the age of the member from youngest to oldest.

Example of output (first 15 records):

MemID	Surname	FirstName	DateOfBirth	E-Mail	Gender
1	Itzak	Dory	2003/04/09	ditzak0@nytimes.com	Male
38	Ruzic	Jennine	2002/10/10	jruzic11@google.it	Female
30	Chezier	Fax	2001/12/08	fcheziert@wsj.com	Male
11	Oxlade	Judah	2000/09/25	joxladea@irs.gov	Male
16	Godier	Ainsley	1999/12/02	agodierf@aol.com	Female
5	Radenhurst	Dare	1999/11/16	dradenhurst4@booking.com	Male
50	Tiler	Ambrosio	1999/03/28	atiler1d@ning.com	Male
31	Yetman	Udell	1997/04/01	uyetmanu@cnet.com	Male
19	Brosini	Domenico	1996/07/09	dbrosinii@cornell.edu	Male
37	Cornborough	Dennet	1996/03/30	dcornborough10@mlb.com	Male
36	Holdron	Farlay	1993/04/09	fholdronz@trellian.com	Male
8	Drains	Cassey	1993/01/07	cdrains7@tmall.com	Female
20	Marrison	Adria	1992/02/20	amarrisonj@printfriendly.com	Female
47	Harbage	Mile	1989/09/03	mharbage1a@miitbeian.gov.cn	Male
14	Carlon	Lyndsie	1988/02/07	lcarlond@google.ca	Female

(3)

2.1.2 Button [Q2.1.2]

Write SQL code to display the various types of Crypto available as well as the total amount of Crypto in each type.

Example of output:

CryptoType	Amount
Bitcoin	58.4
Ethereum	74.1
Litecoin	69.5
Ripple	60.4

(4)

2.1.3 Button [Q2.1.3]

The user would like to know how many members celebrate their birthday in a certain month. Code has been provided to allow the user to fill in the month using an input box.

Write SQL code to display the number of birthdays that will be celebrated in the month chosen in the input box. Name the calculated field "Birthdays in [the correct month]". *You may make use of the provided constant array: ARRMONTHS.*

Example of output:

Input = 1

Birthdays in January
2

Input = 6

Birthdays in June
7

Input = 10

Birthdays in October
3

(6)

2.1.4 **Button [Q2.1.4]**

You would like to see all the members who have Litecoin with the value of over R5 500. A variable with Litecoin's exchange value has been declared and assigned for you.

Write a SQL statement to display the Surname, Firstname, CryptoType, Amount, and the value of the Litecoin for that person. Name the calculated field "Value" and it must be displayed in Rands.

Example of output:

Surname	Firstname	CryptoType	Amount	Value
Lightwood	Gannon	Litecoin	5.5	R7 366.15
Drains	Cassey	Litecoin	6.3	R8 437.59
Lampert	Arel	Litecoin	4.5	R6 026.85
Witsey	Jorge	Litecoin	4.6	R6 160.78
Cornborough	Dennet	Litecoin	4.3	R5 758.99
Tiler	Ambrosio	Litecoin	6.3	R8 437.59

(9)

2.1.5 **Button [Q2.1.5]**

All investors have decided to burn their Ripple wallets.

Write a SQL statement to delete all Ripple entries.

NOTE: *Since all Ripple is now removed, if you click on btn2.1.2, the following will output.*

CryptoType	Amount
Bitcoin	58.4
Ethereum	74.1
Litecoin	69.5

(3)

2.1.6 Button [Q2.1.6]

Bitcoin has started with its Bitcoin halving process.

Write a SQL statement to change all of the Bitcoin amounts to half of their current amount.

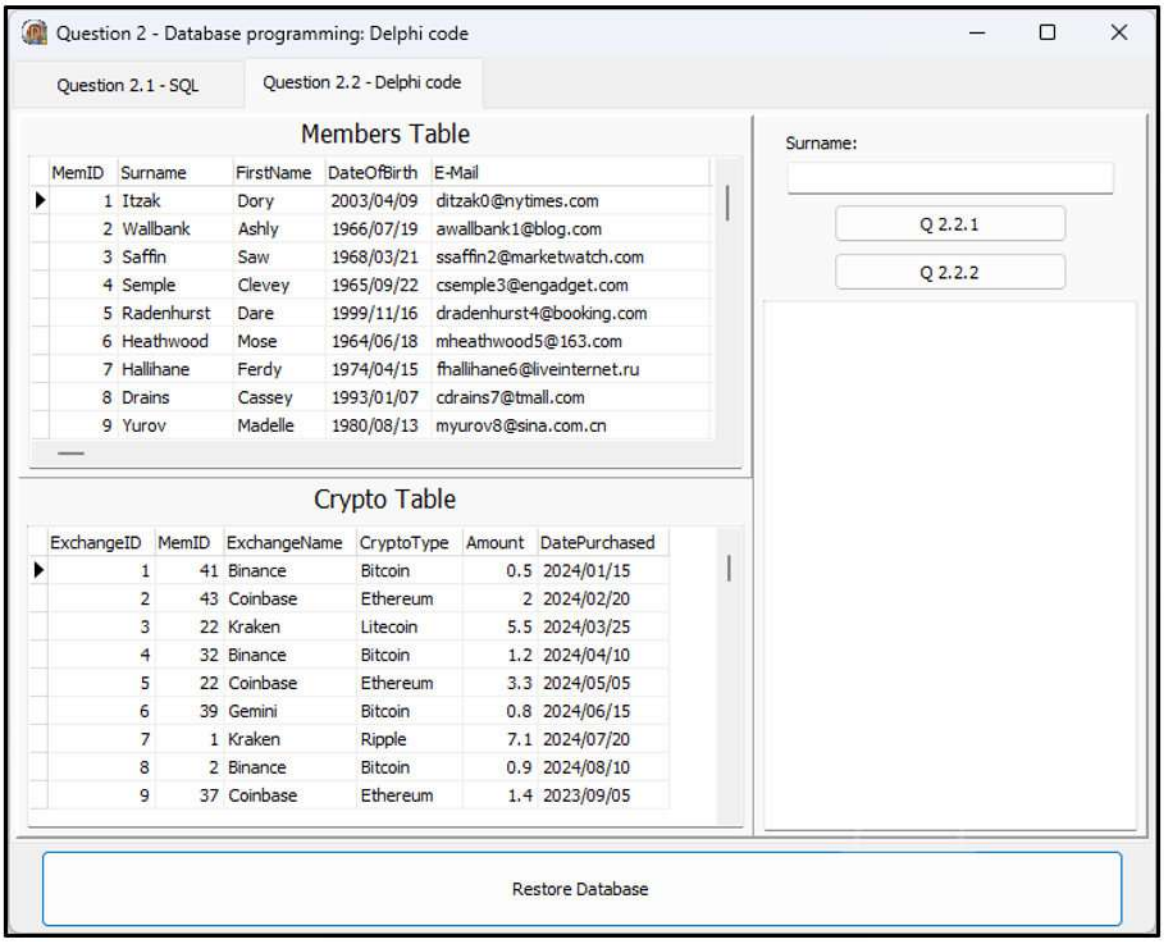
NOTE: *Since Bitcoin has halved in amount, you can check your output using btn2.1.2, the following will output.*

CryptoType	Amount
Bitcoin	29.2
Ethereum	74.1
Litecoin	69.5

(3)

2.2 Tab Sheet [Question 2.2]

Example of the graphical user interface (GUI) for QUESTION 2.2.



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.
- Use the global variables, tblMembers and tblCrypto, provided.

2.2.1 **Button [Q2.2.1]**

Member details need to be displayed when a user searches for a member. Code has been provided to you to extract the surname from the edit box as well as loop through the Members table.

Write code to search in the loop if a member exists. If they exist, output to the rich edit **redOutput** as per example screenshot (Name, Surname, E-Mail, Gender).

Example of output:

If "Lightwood" was entered in the search.

Member's details
Name: Gannon
Surname: Lightwood
E-Mail: glightwoodl@nih.gov
Gender: Male

If "Soap" was entered in the search.

Member's details
Soap was not found in database

(5)

2.2.2 **Button [Q2.2.1]**

Currently 50 members are registered in the Members table. A request was made to display the various genders and how many of each gender owned some form of Cryptocurrency. **NOTE:** There are only FOUR gender types registered.

Code has been provided to you to display the heading Total Members = 50. Write code to loop through the Members table and tally the various genders. Output to the rich edit **redOutput** as per example screenshots. Add a test to determine if your total gender count is the same or different to the original.

You may make use of standard variables or arrays to solve this question.

Example of output:

If all data is correct.

Total members = 50

Male	26
Female	22
Genderfluid	1
Non-binary	1

Correct

If data is incorrect. In this case faulty spelling: Females instead of Female

Total members = 50

Male	26
Females	0
Genderfluid	1
Non-binary	1

Incorrect

(12)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[45]

QUESTION 3: OBJECT-ORIENTED PROGRAMMING

Traders can create a Crypto account and select one of three different Cryptocurrencies: Bitcoin, Ethereum or Litecoin. Once they have created their account, they can purchase their chosen crypto asset by entering a Rand value they are willing to spend. The program will determine the correct amount of the crypto asset based on the exchange rate of the cryptocurrency. Since Cryptocurrencies fluctuate so much, traders can view the current value of their crypto asset in the "Live" viewer. The program updates every second and will display whether the trader has made a Profit or a Loss based on their original purchase value.

Do the following:

- Open the incomplete program in the **Question 3** folder.
- Open the incomplete object class **Crypto_U.pas**.
- Enter your full name as a comment in in the first line of both the **Crypto_U.pas** file and the **Question3_u.pas** file.
- Compile and execute the program. The program has limited functionality currently.
- Do NOT remove or change any provided code.

Example of the graphical user interface (GUI):

Complete the code as specified in QUESTION 3.1 and QUESTION 3.2 that follow.

*****NOTE:** *You are NOT allowed to add any additional attributes or user-defined methods, unless explicitly stated in the question.*****

Open the incomplete object class **Crypto_U.pas.**

3.1 The provided incomplete class (**TCrypto**) contains the declaration of six attributes that describe the **objCrypto** object.

Name of attributes	Description
fFirstName	String value containing trader's first name
fSurname	String value containing trader's surname
fCrypto	Integer value containing which Cryptocurrency the trader selected
fAmount	Real value containing the Cryptocurrency amount
fOriginalValue	Real value containing the money originally spent by the trader
fCurrentValue	Real value containing the current value in Rands of the trader's Cryptocurrency

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

3.1.1 Write code for a **constructor** method named **Create** that will receive the following parameters:

- First name
- Surname
- Cryptocurrency (value ranging between 0 and 2)

Assign these parameter values to the correct attributes and set the remaining attributes to a default value of 0. (5)

3.1.2 Write an accessor method called **getCryptoName** that returns the full name of the cryptocurrency based on the **fCrypto** field attribute. The mapping is as follows:

- 0 = Bitcoin
- 1 = Ethereum
- 2 = Litecoin

Uncomment the code in the toString method. (4)

3.1.3 Write code for a mutator method called **setAmount** that receives a value (purchase value of cryptocurrency) as a parameter.
Set the **fAmount** attribute based on the following rules by multiplying the purchase value (received by the main unit) with the exchange rate:

fCrypto	Conversion rate
0	1 : 1 000 000
1	1 : 50 000
2	1 : 2 500

Exchange Rate Calculation: 1 ÷ 1000000 = 0.000001

Set the **fOriginalValue** attribute to the purchase value received from the main unit. This value must be able to update as more cryptocurrency of the same type is purchased. (7)

3.1.4 Write code for a mutator method called **setValue.** This method will set the **fCurrentValue** parameter to the current exchange of the cryptocurrency (**fAmount** * exchange rate). Since cryptocurrencies are so volatile and constantly fluctuate, the conversion rates will be based on random values between certain ranges.

Set the **fCurrentValue** attribute based on the following rules:

fCrypto	Currency rate is a random value (both values inclusive)
0	400 000 – 1 600 000
1	25 000 – 75 000
2	500 – 3000

(5)

3.1.5 Write a method called **calcProfitLoss** that will return a string value of a 'Profit', 'Loss' or 'Even'.

The method will make use of the object's attributes to determine the output.
- If the current value is greater than the original price, then it is a profit.
- If the current value is less than the original price, then it is a loss.
- If the current value is equal to the original price, then it is even. (6)

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 **objCrypto**.
Write code to perform the tasks described in QUESTION 3.2.1 to QUESTION 3.2.3 below.

3.2.1 **Button [3.2.1 – Instantiate Crypto Object]**

Write code to do the following:
- Extract the name from the edit box **edtName**, the surname from the edit box **edtSurname** and the number from the radio group **rgpCrypto**.
- Use the information to instantiate the new Crypto object.
- Display a message to the user that the Crypto Account has been created. (6)

3.2.2 **Button [3.2.2 - Purchase Crypto]**

The crypto trader will enter their desired purchase price into the edit box.

Write code to do the following:
- Extract the price from the edit box **edtMoney.**
- Call the **setAmount** method using the value from the edit box. (2)

3.2.3 **Timer [tmrLive]**

When the Live button is pressed, the program will update every second and display the current value of the cryptocurrency along with whether the trader has made a profit or a loss compared to their original purchase value.

Write code on the timer **tmrLive** to do the following:
- Call the **setValue** method.
- Use the **toString** method to display the information of the updated Crypto object in the rich edit **redOutput**.
- Use the **calcProfitLoss** method to display the updated Crypto object in the panel **pnlOutput**.

Example of output if Joe Soap bought R50 000 of Ethereum: **Rand values will be random.**

Profit

Loss

Example of output if Lutho Mzi bought R10 000 of Bitcoin:

Profit

Loss

(5)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[40]

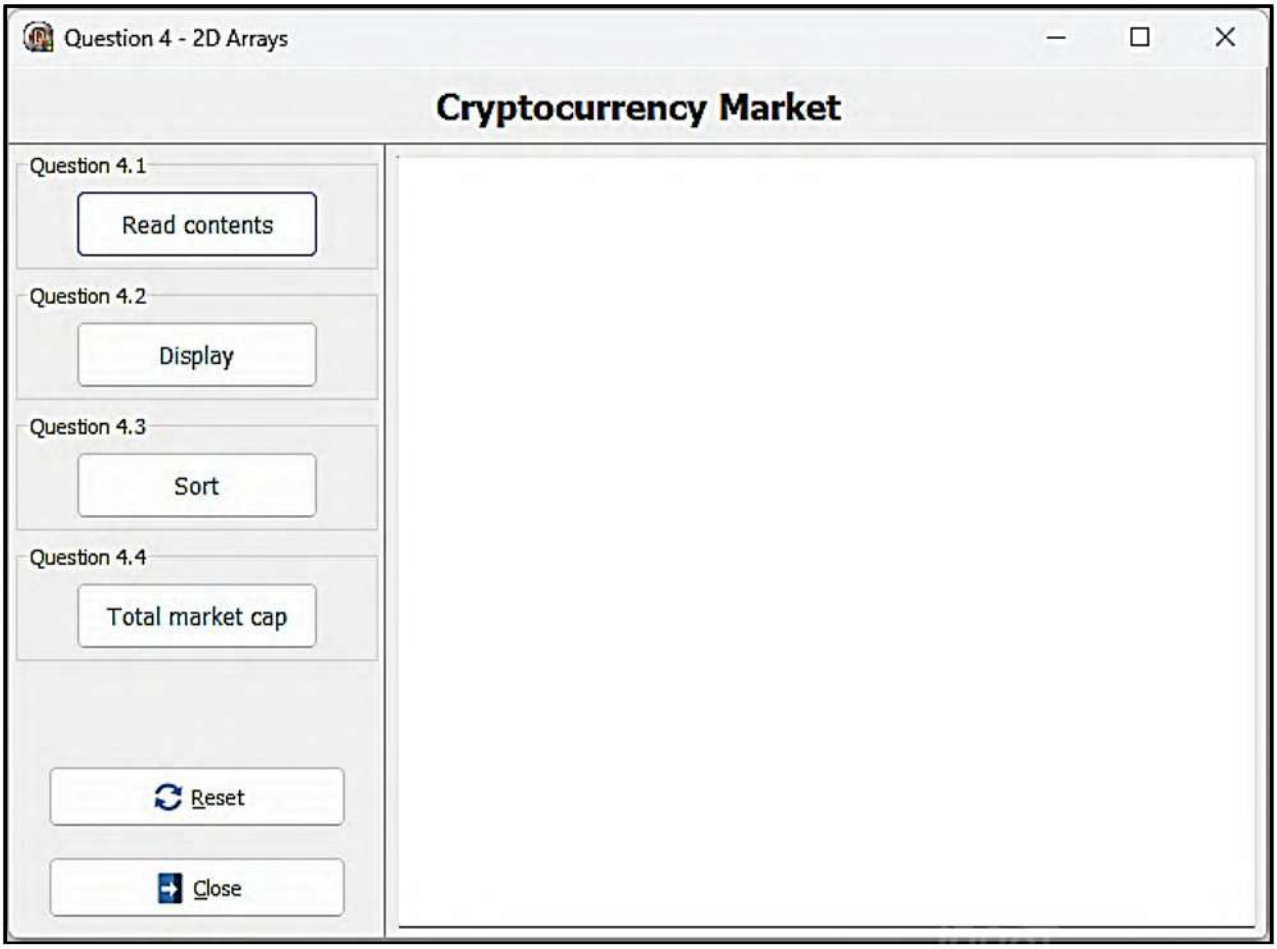
QUESTION 4: PROBLEM-SOLVING PROGRAMMING

The cryptocurrency market consists of trillions of rands worth of money. This application has been created to view 20 popular cryptocurrencies, along with their market capitalisation in USD and currency value in ZAR (South African Rand).

Do the following:

- Open the incomplete program in the **Question 4** folder.
- Enter your full name as a comment in in the first line of the **Question4_u.pas** file.
- Compile and execute the program. The program has limited functionality currently.
- Do NOT remove or change any provided code.

Example of the graphical user interface (GUI):



The following have been provided in the program:

- A text file called Crypto.txt holds data about 20 different cryptocurrencies in the following format:

Cryptocurrency Name, Symbol, Current Price in Rand (ZAR) Market Capitalisation in Dollars (USD).

- A counter variable has been instantiated to store the number of cryptocurrencies from the text file.
- A two-dimensional array called ar2Crypto has been instantiated for you.
- This array will be used to store the data from the text file about the 20 different cryptocurrencies.

Complete the code for each section of QUESTION 4, as described in QUESTION 4.1 to QUESTION 4.4 below.

4.1 **Button [4.1 – Read contents]**

The data from the text file **Crypto.txt** needs to be read into the two-dimensional array **ar2Crypto**. Each row of the array should represent a cryptocurrency and each column should hold the respective data field (Name, Symbol, CurrentPriceInZAR, MarketCapInUSD).

Write code to do the following:

- Read the data from the text file **Crypto.txt** and populate the two-dimensional array **ar2Crypto**.
- Include any necessary error handling for opening the text file. (12)

4.2 **Button [4.2 – Display]**

The contents of the two-dimensional **ar2Crypto** must be displayed neatly in a tabular format.

Write code to do the following:

- Display the contents of the two-dimensional array in the rich edit **redOutput.**
- Ensure that the output is well-aligned.

Example of output:

Name	Symbol	Price (ZAR)	Market Cap (USD)
Avalanche	AVAX	492.66	10762733103
Bitcoin	BTC	1144948.86	1253479481112
Cardano	ADA	6.867	13633857286
Chainlink	LINK	254.34	8273074048
Dogecoin	DOGE	2.205	17731964008
EOS	EOS	10.348	861165571
Ethereum	ETH	62539.38	424724700698
IOTA	IOTA	3.160	580488943
Litecoin	LTC	1309.5	5427650205
Monero	XMR	3039.84	3120443517
NEO	NEO	201.96	790975314
Polkadot	DOT	101.52	8112011565
Ripple	XRP	8.784	27100594642
Solana	SOL	2364.3	60679282963
Stellar	XLM	1.676	2711919206
Tezos	XTZ	14.209	776710653
TRON	TRX	2.098	10172073094
Uniswap	UNI	179.64	5983334594
VeChain	VET	0.481	2168058200
Zcash	ZEC	358.56	325262669

(5)

4.3 **Button [4.3 – Sort]**

Cryptocurrencies are ranked according to their market capitalisation.

Write code to do the following:

- Sort the two-dimensional array based on the current market capitalisation in descending order.
- Display the contents of the sorted two-dimensional array in the rich edit **redOutput.**

Example of output:

Name	Symbol	Price (ZAR)	Market Cap (USD)
Bitcoin	BTC	1144948.86	1253479481112
Ethereum	ETH	62539.38	424724700698
Solana	SOL	2364.3	60679282963
Ripple	XRP	8.784	27100594642
Dogecoin	DOGE	2.205	17731964008
Cardano	ADA	6.867	13633857286
Avalanche	AVAX	492.66	10762733103
TRON	TRX	2.098	10172073094
Chainlink	LINK	254.34	8273074048
Polkadot	DOT	101.52	8112011565
Uniswap	UNI	179.64	5983334594
Litecoin	LTC	1309.5	5427650205
Monero	XMR	3039.84	3120443517
Stellar	XLM	1.676	2711919206
VeChain	VET	0.481	2168058200
EOS	EOS	10.348	861165571
NEO	NEO	201.96	790975314
Tezos	XTZ	14.209	776710653
IOTA	IOTA	3.160	580488943
Zcash	ZEC	358.56	325262669

(10)

4.4 **Button [4.3 – Total market cap]**

The total market capitalisation is the sum of the MarketCapInUSD values for all cryptocurrencies stored in the two-dimensional array.

Write code to do the following:

- Calculate the total market capitalisation and convert it to a rand (ZAR) value. The current exchange rate is:
  1 USD = 18 ZAR
- Display the total market capitalisation in the rich edit **redOutput.**

Example of output:

```
Total Market Cap: R33 433 484 065 038.00
```

(3)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[30]

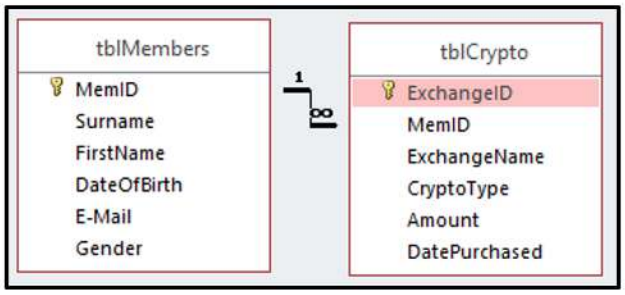
TOTAL: 150

INFORMATION TECHNOLOGY P1

DATABASE INFORMATION QUESTION 2:

The database **CryptoExchange** consists of tables **tblMembers** and **tblCrypto**.

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



The design of the database tables is as follows:

Table: **tblMembers** – This table contains details of the crypto trader members.

Field Name	Data Type	Description
MemID	Number	Unique ID for the crypto trader
Surname	Text (30)	The surname of the crypto trader
FirstName	Text (20)	The name of the crypto trader
DateOfBirth	Date/Time	The date of birth of the crypto trader
E-Mail	Text (255)	The email address of the crypto trader
Gender	Text (20)	The gender of the crypto trader

Example of the first few records in the **tblMembers** table:

MemID	Surname	FirstName	DateOfBirth	E-Mail	Gender
1	Itzak	Dory	2003/04/09	ditzak0@nytimes.com	Male
2	Wallbank	Ashly	1966/07/19	awallbank1@blog.com	Female
3	Saffin	Saw	1968/03/21	ssaffin2@marketwatch.com	Male
4	Semple	Clevey	1965/09/22	csemple3@engadget.com	Male
5	Radenhurst	Dare	1999/11/16	dradenhurst4@booking.com	Male
6	Heathwood	Mose	1964/06/18	mheathwood5@163.com	Male
7	Hallihane	Ferdy	1974/04/15	fhallihane6@liveinternet.ru	Male
8	Drains	Cassey	1993/01/07	cdrains7@tmall.com	Female
9	Yurov	Madelle	1980/08/13	myurov8@sina.com.cn	Female
10	Acott	Reine	1957/06/22	racott9@lulu.com	Female
11	Oxlade	Judah	2000/09/25	joxladea@irs.gov	Male
12	Dawks	Monte	1984/06/14	mdawksb@t-online.de	Male
13	Matelaitis	Neal	1987/08/18	nmatelaitisc@arizona.edu	Male
14	Carlon	Lyndsie	1988/02/07	lcarlond@google.ca	Female
15	Glamart	Carroll	1954/09/01	cglamarte@geocities.com	Male
16	Godier	Ainsley	1999/12/02	agodierf@aol.com	Female
17	Stailey	Jeniffer	1967/12/19	jstaileyg@networksolutions.com	Female
18	Rookes	Wilow	1977/06/27	wrookesh@engadget.com	Female
19	Brosini	Domenico	1996/07/09	dbrosinii@cornell.edu	Male
20	Marrison	Adria	1992/02/20	amarrisonj@printfriendly.com	Female
21	Frisdick	Abran	1957/11/22	afrisdickk@dot.gov	Male
22	Lightwood	Gannon	1971/01/21	glightwoodl@nih.gov	Male

Table: **tblCrypto** – This table contains details of the crypto transactions.

Field Name	Data Type	Description
ExchangeID	Number	Unique ID for the crypto trader
MemID	Number	The ID for the crypto trader who made crypto transaction
ExchangeName	Text (20)	The crypto exchange name
CryptoType	Text (30)	The crypto type
Amount	Number	The amount of crypto purchased in the transaction
DatePurchased	Date/Time	Date the purchase transaction was made

Example of the first few records in the **tblCrypto** table:

ExchangeID	MemID	ExchangeName	CryptoType	Amount	DatePurchased
1	41	Binance	Bitcoin	0.5	2024/01/15
2	43	Coinbase	Ethereum	2	2024/02/20
3	22	Kraken	Litecoin	5.5	2024/03/25
4	32	Binance	Bitcoin	1.2	2024/04/10
5	22	Coinbase	Ethereum	3.3	2024/05/05
6	39	Gemini	Bitcoin	0.8	2024/06/15
7	1	Kraken	Ripple	7.1	2024/07/20
8	2	Binance	Bitcoin	0.9	2024/08/10
9	37	Coinbase	Ethereum	1.4	2023/09/05
10	8	Gemini	Litecoin	6.3	2023/10/12
11	32	Binance	Bitcoin	2.5	2023/11/15
12	36	Coinbase	Ethereum	3.6	2021/12/20
13	24	Kraken	Litecoin	4.5	2020/01/18
14	30	Gemini	Ripple	1.1	2024/02/22
15	14	Binance	Bitcoin	3.2	2024/03/27
16	12	Coinbase	Ethereum	4	2024/04/11
17	48	Kraken	Litecoin	2.2	2024/05/06
18	41	Gemini	Ripple	5.6	2024/06/16
19	47	Binance	Bitcoin	1.7	2024/07/21
20	16	Coinbase	Ethereum	2.9	2023/08/11
21	24	Kraken	Litecoin	3.8	2022/09/06
22	25	Gemini	Ripple	0.5	2023/10/13
23	43	Binance	Bitcoin	0.6	2023/11/16
24	44	Coinbase	Ethereum	1.3	2022/12/21
25	47	Kraken	Litecoin	2.7	2024/01/19
26	26	Gemini	Ripple	4.1	2024/02/23
27	12	Binance	Bitcoin	1.8	2024/03/28
28	15	Coinbase	Ethereum	2.5	2024/04/12
29	15	Kraken	Litecoin	0.9	2024/05/07
30	15	Gemini	Ripple	3	2024/06/17

Province of the
EASTERN CAPE
EDUCATION

Iphondo leMpuma Kapa: Isebe leMfundo
Provinsie van die Oos Kaap: Department van Onderwys
Porafensie Ya Kapa Botjhabela: Lefapha la Thuto

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2024

INFORMATION TECHNOLOGY P1 MARKING GUIDELINE

MARKS: **150**

This marking guideline consists of 21 pages.

NAME OF LEARNER:				
TOTAL QUESTION 1	**TOTAL QUESTION 2**	**TOTAL QUESTION 3**	**TOTAL QUESTION 4**	**TOTAL**
/35	**/45**	**/40**	**/30**	**/150**

	QUESTION 1	MAX MARK	MARKS ACHIEVED
1.1	**BUTTON: [Q1.1 – Panel Properties]** Set the font type to Arial. ✓ Set the font size to 16. ✓ Set the font to Bold ✓ and Italic. ✓ Set the text to display on the right-hand side of the panel. ✓ Change the caption to 'Crypto Exchange'. ✓	**6**	
1.2	**BUTTON: [Q1.2 – Share Coins]** Get input from edit box ✓ Calculate shares using the total of 12 (3+4+5) ✓ Calculation for each person (share/12*input from edit box) ✓ Use DIV/trunc/floor to discard decimals ✓ Calculate Remaining Coins ✓ Display in the label Tom's Share ✓ Jerry's Share ✓ Andile's Share ✓ Remaining Coins ✓ Correct use of enter space and display of apostrophes after each person's name ✓	**10**	
1.3	**BUTTON: [Q1.3 – Password]** Initialise Password String ✓ Loop for each letter in "CRYPTO" (6 times) ✓ Initialise Number variable to 0 ✓ Conditional Loop with correct condition ✓ Increment Number variable ✓ Randomise a number ✓ in correct range ✓ Get the corresponding letter for the randomised number ✓ Build password string: ✓ include the number ✓ and the lowercase ✓ letter of the character. Display the Password in the label. ✓	**12**	

1.4	**BUTTON: [Q1.4 – ASCII Art]**
	
	Get the length of the longest line and assign to iLines ✓
	
	Initialise an empty string sLine ✓
	
	// Constructing the pattern by adding '&' characters
	Loop ✓
	Add '&' character to sLine ✓
	Display sLine in rich edit ✓
	
	// Removing '&' characters from the beginning of each line
	Loop
	Remove the first character from sLine/Build a string ✓
	Display sLine in rich edit ✓
		7	

	TOTAL QUESTION 1	**35**	

QUESTION 2		MAX. MARK	MARKS ACHIEVED
2.1.1	**Button [Q2.1.1]** 'SELECT * FROM tblMembers ORDER BY DateOfBirth DESC' SELECT * (all fields) ✓ FROM correct table ✓ ORDER BY correct field DESC ✓	3	
2.1.2	**Button [Q2.1.2]** 'SELECT CyptoType, Sum(Amount) AS [Amount] FROM tblCrypto GROUP BY CryptoType' SELECT CryptoType ✓, Sum(correct field) ✓ FROM correct table ✓ GROUP BY correct field ✓	4	
2.1.3	**Button [Q2.1.3]** 'SELECT count(DateOfBirth) AS [Birthdays in ' + arrMonths[StrToInt(sLine)] + '] FROM tblMembers WHERE Month(DateOfBirth) = ' + sLine SELECT count(correct field) ✓AS [Birthdays in correct month ✓] ✓ FROM correct table ✓ WHERE Month(correct field) ✓ = input variable ✓	6	
2.1.4	**Button [Q2.1.4]** 'SELECT Surname, Firstname, CryptoType, Amount, format(Amount * ' + FloatToStr(rLitecoin) + ',"Currency") AS [Value] FROM tblMembers, tblCrypto WHERE tblMembers.MemID = tblCrypto.MemID AND CryptoType = "Litecoin" AND (Amount * ' + FloatToStr(rLitecoin) +') > 5500' SELECT Surname, Firstname, CryptoType, Amount ✓ , Format (Amount * LiteCoin Value ✓, "Currency" ✓) AS [Value] ✓, FROM both tables ✓ (tblMembers, tblCrypto) WHERE link between tables ✓ (tblMembers.MemID = tblCrypto.MemID) AND CryptoType = "Litecoin" ✓AND (Amount * Litecoin value) ✓ > 5500' ✓	9	
2.1.5	**Button [Q2.1.5]** 'DELETE FROM tblCrypto WHERE CryptoType = "Ripple" ' DELETE FROM correct table ✓ WHERE correct field ✓ = "Ripple" ✓	3	
2.1.6	**Button [Q2.1.6]** 'UPDATE tblCrypto SET Amount = Amount / 2 WHERE CryptoType = "Bitcoin" ' UPDATE correct table ✓ SET Amount = Amount / 2 ✓ WHERE CryptoType = "Bitcoin" ✓	3	

2.2.1	**Button [Q2.2.1]** Test if valid condition: correct Surname ✓ = data field value ✓ Display the correct fields (Name; Surname; E-Mail; Gender) ✓ in the correct format ✓ Move to next record in table ✓	**5**	
2.2.2	**Button [Q2.2.2]** Instantiate ✓ and Initialise ✓ for Counting and Gender Types (standard variables or arrays) Test if valid condition: correct Gender = data field value ✓ Increment appropriate Counter ✓ Increment Total ✓ Move to next record in table ✓ Test if valid condition: Total = record count of Members table ✓ String Variable ← Correct ✓ else String Variable ← Incorrect ✓ Display the correct fields (Male, Female, Genderfluid, Non-binary) in the correct format ✓✓ Display String variable correct or incorrect ✓	**12**	
	TOTAL QUESTION 2	**45**	

	QUESTION 3	MAX. MARK	MARKS ACHIEVED
3.1.1	**Constructor Create** Constructor definition with three parameters ✓ of correct data type ✓ Assign three parameters to attributes ✓✓ Set remaining attributes to zero ✓	**5**	
3.1.2	**Accessor Method – getCryptoName** Function definition with String return type ✓ If/Case statement ✓ Correct conversion of attribute fCrypto to cryptocurrency name ✓ Return the cryptocurrency name ✓	**4**	
3.1.3	**Mutator Method – setAmount** Procedure definition ✓ with Real parameter value ✓ Determine the exchange rate based on rules (any method): Bitcoin = 0.000001 *1 / 1000000* (given) Ethereum = 0.00002 ✓ *1 / 50000* Litecoin = 0.0004 ✓ *1 / 2500* Assign fAmount to the received parameter * exchange rate ✓ Assign fOriginalValue to fOriginalValue ✓ + received parameter ✓	**7**	
3.1.4	**Mutator Method – setValue** Procedure definition ✓ Determine the exchange rate based on rules (any method): Bitcoin = Random: Range 400 000 – 1 600 000 (both inclusive) ✓ Ethereum = Random: Range 25 000 – 75 000 (both inclusive) ✓ Litecoin = Random: Range 500 – 3000 (both inclusive) ✓ Assign fCurrentValue to fAmount * random exchange rate ✓	**5**	
3.1.5	**Auxiliary Method – calcProfitLoss** Function definition with String return type ✓ Test if (fCurrentValue > fOriginalValue) ✓ Return "Profit" ✓ Else Test if (fCurrentValue < fOriginalValue) ✓ Return "Loss" ✓ Else Return "Even" ✓	**6**	
	Subtotal: Object class	**[27]**	

3.2.1	**Button [3.2.1 - Instantiate Crypto Object]** Retrieve input from components ✓ Instantiate object: objCrypto := ✓ TCrypto.Create ✓ parameters of correct data type ✓ and order ✓ Display message indicating object was instantiated ✓	**6**	
3.2.2	**Button [3.2.2 – Purchase Crypto]** Extract the price from the edit box ✓ Use setAmount method with price argument ✓ to set attribute	**2**	
3.2.3	**Timer [tmrLive]** Use setValue method ✓ to set attribute Display object in rich edit ✓ using toString method ✓ Display Profit/Loss in panel ✓ using calcProfitLoss method ✓	**5**	
	Subtotal: Main unit	**[13]**	
	TOTAL QUESTION 3	**40**	

	QUESTION 4	MAX. MARK	MARKS ACHIEVED
4.1	**Button [4.1 – Read contents]** Assign the text file ✓ Test if file exists ✓ If not, show a message ✓ Reset the text file ✓ Initialise counter to 0 ✓ While loop through the text file ✓ Increment the counter ✓ Read a value from text file into string variable ✓ Loop columns from 1 to 3 ✓ Find position of comma in string variable ✓ Assign string variable to correct 2D Array[Counter,Column] ✓ Delete out of string variable up to and including the comma Assign remaining string variable to 2D Array[Counter,Column 4] ✓	**12**	
4.2	**Button [4.2 – Display]** Loop Row from 1 to Counter ✓ Loop Col from 1 to 4 ✓ String variable assigned string variable + 2D Array[Row,Col] ✓ + Tab space ✓ Display in the richedit ✓	**5**	
4.3	**Button [4.3 – Sort]** Loop Outer from 1 to Counter - 1 ✓ Loop Inner from Outer + 1 to Counter ✓ Test if 2D Array(Inner,4) ✓ is less than 2D Array(Outer,4) ✓ *must convert string column 4 to a number* ✓ Loop Col 1 to 4 ✓ String Temp ← 2D Array[Outer,Col] ✓ 2D Array[Outer,Col] ← 2D Array[Inner,Col] ✓ 2D Array[Inner,Col] ← String Temp ✓ Display sorted 2D Array in rich edit ✓	**10**	
4.4	**Button [4.4 – Total market cap]** Variable Total must be either Real or Int64 Initialise Total to 0 Loop Rows from 1 to Counter ✓ Total assigned Total + (2D Array[Row,Col 4] * 18) ✓ Display Total in rich edit in correct format ✓	**3**	
	TOTAL QUESTION 4	**30**	

SAMPLE SOLUTIONS

QUESTION 1

////////// 35 marks //////////

```
// ========================================================
//                    Question 1.1 – 6 marks
// ========================================================
procedure TfrmQuestion1.btn1_1Click(Sender: TObject);
begin
   /// Enter your code below ///

   with pnlOutput do
      begin
         Font.Name := 'Arial';
         Font.Size := 16;
         Font.Style := [fsBold, fsItalic];
         Alignment := taRightJustify;
         Caption := 'Crypto Exchange';
      end;
   end;


// ========================================================
//                    Question 1.2 – 10 Marks
// ========================================================
procedure TfrmQuestion1.btn1_2Click(Sender: TObject);
var
   iNumCoins : Integer;
   iTom, iJerry, iAndile : Integer;
   iLeftOver : Integer;
begin
   /// Enter your code below ///

   iNumCoins := StrToInt(edtInput.Text);
   iTom      := iNumCoins * 3 DIV 12;
   iJerry    := iNumCoins * 4 DIV 12;
   iAndile   := iNumCoins * 5 DIV 12;
   iLeftOver := iNumCoins – iTom – iJerry - iAndile;

   lbl1_2.Caption := 'Tom"s Share: ' + IntToStr(iTom) + #13 +
                     'Jerry"s Share: ' + IntToStr(iJerry) + #13 +
                     'Andile"s Share: ' + IntToStr(iAndile) + #13 +
                     'Remaining Coins: ' + IntToStr(iLeftOver);
end;
```



```
// ============================================================
//                        Question 1.3 – 12 marks
// ============================================================
procedure TfrmQuestion1.btn1_3Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
CONST
   PASSWORD = 'CRYPTO';
var
   sPassword : String;
   sChar : Char;
   iNum : Integer;
begin
   /// Enter your code below ///

   sPassword := '';
   for var I := 1 to Length(PASSWORD) do
      begin
         iNum := 0;
         repeat
               inc(iNum);
               sChar := Chr(Random(26) + 65);
         until sChar = PASSWORD[I];
         sPassword := sPassword + IntToStr(iNum) + Lowercase(sChar);
      end;
   lbl1_3.Caption := sPassword;
end;

// ============================================================
//                        Question 1.4 – 7 Marks
// ============================================================
procedure TfrmQuestion1.btn1_4Click(Sender: TObject);
var
   sLine : String;
   iLines : Integer;
begin
   /// Enter your code below ///

   redOutput.Clear;
   sLine := '';
   iLines := StrToInt(InputBox('Lines','Enter the number of lines.',''));
   for var I := 1 to iLines do
      begin
         sLine := sLine + '&';
         redOutput.Lines.Add(sLine);
      end;
   for var I := 1 to iLines - 1 do
      begin
         delete(sLine,1,1);
         redOutput.Lines.Add(sLine);
      end;
end;
```


QUESTION 2

////////// 45 marks //////////

```
// ============================================================
//                    Question 2.1.1 – 3 marks
// ============================================================
procedure TfrmQuestion2.btn2_1_1Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
var
   sSQL1: String;
begin
   /// Enter your code below ///

   sSQL1 :=  'SELECT * ' +
             'FROM tblMembers ' +
             'ORDER BY DateOfBirth ASC';

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.runSQL(sSQL1);
   if length(sSQL1) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;


// ============================================================
//                    Question 2.1.2 – 4 Marks
// ============================================================
procedure TfrmQuestion2.btn2_1_2Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
var
   sSQL2: String;
begin
   /// Enter your code below ///

   sSQL2 :=  'SELECT CryptoType, Sum(Amount) AS [Amount] ' +
             'FROM tblCrypto ' +
             'GROUP BY CryptoType';

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.runSQL(sSQL2);
   if length(sSQL2) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;
```



```
// ============================================================
//                        Question 2.1.3 – 6 marks
// ============================================================
procedure TfrmQuestion2.btn2_1_3Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
CONST
   ARRMONTHS : array[1..12] of String = ('January','February','March','April',
                                         'May','June','July','August','September',
                                         'October','November','December');
var
   sSQL3 : String;
   sLine : String;
begin
   sLine := inputbox('Month','Enter your month (1-12)','1');
   /// Enter your code below ///
            //Alternate solution
         //Case statement or If statement to determine month in string format

   sSQL3 :=  'SELECT count(*) AS [Birthdays in ' + ARRMONTHS[StrToInt(sLine)] +'] ' +
             'FROM tblMembers ' +
             'WHERE Month(DateOfBirth) = ' + sLine;

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.runSQL(sSQL3);
   if length(sSQL3) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;

// ============================================================
//                        Question 2.1.4 – 9 Marks
// ============================================================
procedure TfrmQuestion2.btn2_1_4Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
var
   sSQL4: String;
   rLiteCoin : Real;
begin
   rLitecoin := 1339.30;
   /// Enter your code below ///

   sSQL4 :=  'SELECT Surname, Firstname, CryptoType, Amount, ' +
                    'format(Amount * ' + FloatToStr(rLitecoin) + ',"Currency") AS [Value] ' +
             'FROM tblMembers, tblCrypto ' +
             'WHERE tblMembers.MemID = tblCrypto.MemID ' +
             'AND CryptoType = "Litecoin" ' +
             'AND (Amount * ' + FloatToStr(rLitecoin) +') > 5500';

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.runSQL(sSQL4);
   if length(sSQL4) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;
```

```
// ============================================================
//                        Question 2.1.5 – 3 marks
// ============================================================
procedure TfrmQuestion2.btn2_1_5Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
var
   sSQL5: String;
begin
   /// Enter your code below ///

   sSQL5 :=  'DELETE FROM tblCrypto ' +
             'WHERE CryptoType = "Ripple" ';

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.executeSQL(sSQL5,dbgMembers,dbgCrypto,dbgSQL);
   if length(sSQL5) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;


// ============================================================
//                        Question 2.1.6 – 3 Marks
// ============================================================
procedure TfrmQuestion2.btn2_1_6Click(Sender: TObject);
   // Provided code - DO NOT DELETE OR ALTER //
var
   sSQL6: String;
begin
   /// Enter your code below ///

   sSQL6 :=  'UPDATE tblCrypto ' +
             'SET Amount = Amount / 2 ' +
             'WHERE CryptoType = "Bitcoin"';

   // Provided code - DO NOT DELETE OR ALTER //
   dbCONN.executeSQL(sSQL6,dbgMembers,dbgCrypto,dbgSQL);
   if length(sSQL6) <> 0 then
      SetGridColumnWidths(dbgSQL);
end;
```



```
// ============================================================
//                        Question 2.2.1 – 5 marks
// ============================================================
procedure TfrmQuestion2.btn2_2_1Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
   sSurname : String;
   sString : String;
begin
   with redOutput do
      begin
         Clear;
         SelAttributes.Style := [fsBold];
         Lines.Add('Member''s details');
      end;
   with tblMembers do
      begin
         Open;
         First;
         sSurname := edtSurname.Text;
         sString := sSurname + ' was not found in database';
         while not (eof) do
            begin
               /// Enter your code below ///

               if UpperCase(FieldByName('Surname').AsString) = UpperCase(sSurname) then
                  begin
                        sString := 'Name: ' + FieldByName('FirstName').AsString + #13 +
                                   'Surname: ' + FieldByName('Surname').AsString + #13 +
                                   'E-Mail: ' + FieldByName('E-mail').AsString + #13 +
                                   'Gender: ' + FieldByName('Gender').AsString;
                  end;
               Next;
            end;
         redOutput.Lines.Add(sString);
      end;
end;
```



```
// ============================================================
//                      Question 2.2.2 – 12 Marks
// ============================================================
procedure TfrmQuestion2.btn2_2_2Click(Sender: TObject);
var
   sGender, sLine : String;
   arrGender : Array[1..4] of String;
   arrCryptoCount : Array[1..4] of Integer;
   iTotal, K : Integer;
begin
   // Provided code - DO NOT DELETE OR ALTER //
   with redOutput do
      begin
         Clear;
         SelAttributes.Style := [fsBold];
         Lines.Add('Total members = ' + IntToStr(tblMembers.RecordCount));
         Lines.Add('-------------------------');
      end;
   with tblMembers do
      begin
         Open;
         First;
         /// Enter your code below ///

         arrGender[1] := 'Male';
         arrGender[2] := 'Female';
         arrGender[3] := 'Genderfluid';
         arrGender[4] := 'Non-binary';
         iTotal := 0;
         for K := 1 to 4 do
               arrCryptoCount[K] := 0;

         while not (eof) do
            begin
               for K := 1 to 4 do
                  begin
                     if FieldByName('Gender').AsString = arrGender[K] then
                       begin
                            inc(arrCryptoCount[K]);
                            inc(iTotal);
                       end;
            end;
         Next;
       end;
      if iTotal = RecordCount then
         sLine := 'Correct'
      else
         sLine := 'Incorrect';
      for K := 1 to 4 do
         redOutput.Lines.Add(arrGender[K] + #9 + IntToStr(arrCryptoCount[K]));
      redOutput.Lines.Add(#13 + sLine);
   end;
end;
```

QUESTION 3

////////// 40 marks //////////

```
// ============================================================
//                    Question 3.1.1 – 5 marks
// ============================================================
constructor TCrypto.create(sFirstName, sSurname: String; iCrypto: Integer);
begin
    fFirstName      := sFirstName;
    fSurname        := sSurname;
    fCrypto         := iCrypto;
    fAmount         := 0;
    fOriginalValue  := 0;
    fCurrentValue   := 0;
end;


// ============================================================
//                    Question 3.1.2 – 4 Marks
// ============================================================
function TCrypto.getCryptoName: String;
begin
    case fCrypto of
        0 : Result := 'Bitcoin';
        1 : Result := 'Ethereum';
        2 : Result := 'Litecoin';
    end;
end;


// ============================================================
//                    Question 3.1.3 – 7 Marks
// ============================================================
procedure TCrypto.setAmount(rMoney : Real);
var
    rConversion : Real;
begin
    rConversion := 0;
    case fCrypto of
        0 : rConversion := 1 / 1000000;
        1 : rConversion := 1 / 50000;
        2 : rConversion := 1 / 2500;
    end;
    fAmount := rMoney * rConversion;
    fOriginalValue := fOriginalValue + rMoney;
end;
```



```
// ============================================================
//                          Question 3.1.4 – 5 marks
// ============================================================
procedure TCrypto.setValue;
begin
    case fCrypto of
        0 : fCurrentValue := fAmount * randomRange(400000,1600001);
        1 : fCurrentValue := fAmount * randomRange(25000,75001);
        2 : fCurrentValue := fAmount * randomRange(500,3001);
    end;
end;


// ============================================================
//                          Question 3.1.5 – 6 Marks
// ============================================================
function TCrypto.calcProfitLoss: String;
begin
    if fCurrentValue > fOriginalValue then
       Result := 'Profit'
    else
       if fCurrentValue < fOriginalValue then
              Result := 'Loss'
       else
              Result := 'Even';
end;
```

```
// ============================================================
//                        Question 3.2.1 – 6 marks
// ============================================================
procedure TfrmQuestion3.btn3_2_1Click(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto := TCrypto.Create(edtName.Text, edtSurname.Text, rgpCrypto.ItemIndex);
    ShowMessage('Crypto account created successfully');

    // Provided code - DO NOT DELETE OR ALTER //
    btn3_2_2.Enabled := True;
end;


// ============================================================
//                        Question 3.2.2 – 2 Marks
// ============================================================
procedure TfrmQuestion3.btn3_2_2Click(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto.setAmount(StrToFloat(edtMoney.Text));

    // Provided code - DO NOT DELETE OR ALTER //
    btnLive.Enabled := True;
end;


// ============================================================
//                        Question 3.2.3 – 5 Marks
// ============================================================
procedure TfrmQuestion3.tmrLiveTimer(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto.setValue;
    redOutput.Lines.Add(objCrypto.toString);
    pnlOutput.Caption := objCrypto.calcProfitLoss;
end;
```


QUESTION 4

////////// 30 marks //////////

```
// ============================================================
//                    Question 4.1 – 12 marks
// ============================================================
procedure TfrmQuestion4.btn4_1Click(Sender: TObject);
var
   MyFile : TextFile;
   sLine  : String;
   iPos   : Integer;
   iCol   : Integer;
begin
   /// Enter your code below ///
   AssignFile(MyFile, 'Crypto.txt');
   try
      Reset(MyFile);
   except
      ShowMessage('File not found');
      Exit;
   end;

   iCount := 0;
   while not eof(MyFile) do
      begin
         inc(iCount);
         ReadLn(MyFile, sLine);
         for iCol := 1 to 3 do
            begin
               iPos := pos(',',sLine);
               ar2Crypto[iCount,iCol] := copy(sLine,1,iPos - 1);
               delete(sLine,1,iPos);
            end;
         ar2Crypto[iCount,4] := sLine;
      end;

   CloseFile(MyFile);
end;
```

```
// ============================================================
//                          Question 4.2 – 5 marks
// ============================================================
procedure TfrmQuestion4.btn4_2Click(Sender: TObject);
var
   iRow, iCol : Integer;
   sLine : String;
begin
   // Provided code - DO NOT DELETE OR ALTER //
   with redOutput do
      begin
         Clear;
         Paragraph.TabCount := 3;
         Paragraph.Tab[0] := 60;
         Paragraph.Tab[1] := 120;
         Paragraph.Tab[2] := 200;
         SelAttributes.Style := [fsBold];
         Lines.Add('Name' + #9 + 'Symbol' + #9 + 'Price (ZAR)' + #9 + 'Market Cap (USD)');
      end;
      /// Enter your code below ///
      for iRow := 1 to iCount do
         begin
            sLine := '';
            for iCol := 1 to 4 do
               sLine := sLine + (ar2Crypto[iRow,iCol] + #9);
            redOutput.Lines.Add(sLine);
         end;
end;

// ============================================================
//                          Question 4.3 – 10 Marks
// ============================================================
procedure TfrmQuestion4.btn4_3Click(Sender: TObject);
var
   K, L, J  : Integer;
   sTemp : String;
begin
   /// Enter your code below ///
   for K := 1 to iCount - 1 do
      for L := K + 1 to iCount do
         begin
            if StrToFloat(ar2Crypto[K,4]) < StrToFloat(ar2Crypto[L,4]) then
               begin
                  for Col := 1 to 4 do
                     begin
                       sTemp := ar2Crypto[K,Col];
                       ar2Crypto[K,Col] := ar2Crypto[L,Col];
                       ar2Crypto[L,Col] := sTemp;
                     end;
               end;
         end;
   btn4_2.Click;
end;
```

```
// ============================================================
//                        Question 4.4 – 3 marks
// ============================================================
procedure TfrmQuestion4.btn4_4Click(Sender: TObject);
var
   rTotalCap : Real;
   iRow : Integer;
begin
   /// Enter your code below ///
   rTotalCap := 0;
   for iRow := 1 to iCount do
      rTotalCap := rTotalCap + StrToFloat(ar2Crypto[iRow,4]);
   redOutput.Lines.Add(#13 + 'Total Market Cap: ' +
                        (FloatToStrF(rTotalCap,ffCurrency,25,2) * 18));
end;
```