INFORMATION TECHNOLOGY P1

GRADE 12

PREPARATORY EXAMINATIONS

SEPTEMBER 2023

MARKS: 150

EXAMINER(S): F Ismail
G Surtees
C Andrew
M Gebreab

TIME: 3 hours

MODERATOR: L Roopnundh

This question paper consists of 24 pages.

INSTRUCTIONS AND INFORMATION

1. This paper is divided into FOUR sections. Candidates must answer ALL 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 paper is set in 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 only.

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

9. Make sure that your name appears as a comment in every program that you code.

10. 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.

11. At the end of this examination session you must hand in a disk/CD/DVD/flash disc 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.

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

Do the following:

- Double click on the following password protected executable file: **Trials 2023 DATA.exe**
- Click on the 'Extract' button
- Enter the password : **TriAL$23**

Once extracted the following list of files will be available in the folder Trials 2023 DATA:

Question 1:
frmQuestion1_p.dpr
frmQuestion1_p.dproj
frmQuestion1_p.res
frmQuestion1_U.dfm
frmQuestion1_U.pas
image1.jpg
IronManHand.ico

Question 2:
Heroes.mdb
Heroesbackup.mdb
ConnectDB_u.pas
Question2_p.dpr
Question2_p.dproj
Question2_p.res
Question2_U.dfm
Question2_U.pas

Question 3:
Exhibitors.txt
clsRegistration_u.pas
Question3_p.dpr
Question3_p.dproj
Question3_p.res
Question3_u.dfm
Question3_u.pas

Question 4:
Fixtures.txt
HSEL_logo.png
Q4.dfm
Q4.pas
Quest4_p.dpr
Quest4_p.dproj
Quest4_p.res
Results.txt
Results - Backup

SCENARIO

Comic Con Africa is a Pop Culture and Gaming Festival held every year. It is a platform for local and international entertainment and Pop Culture and it includes some of the world's biggest brands in film, gaming, comics and collectibles.

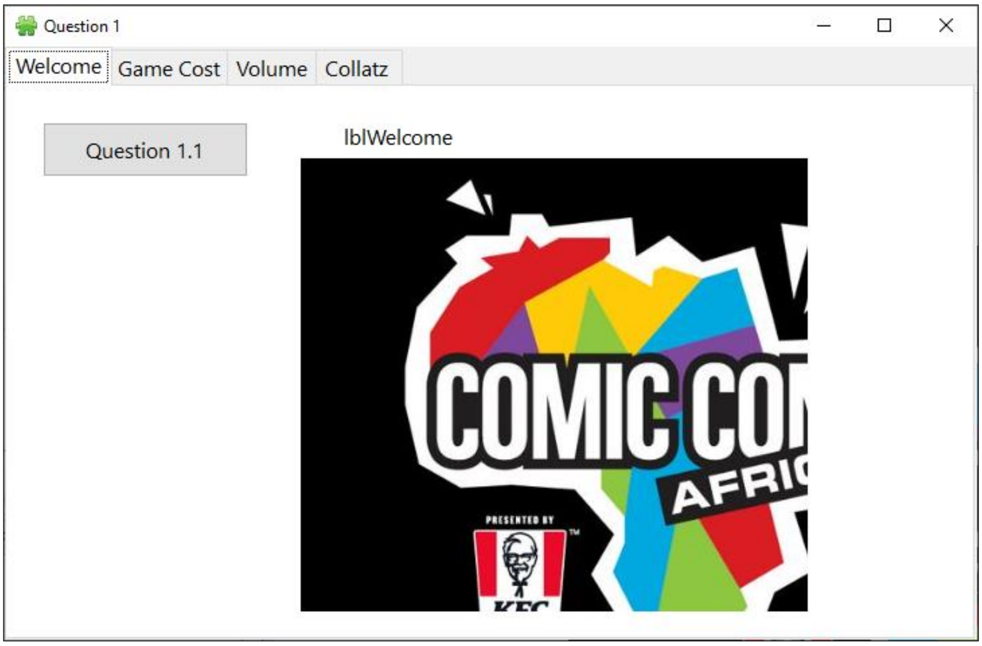
SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

- Open the incomplete program in the folder **Question1**.
- Enter your name and surname as a comment in the first line of the **frmQuestion1_U.pas** file.
- Compile and execute the program. The user interface displays FOUR tab sheets.
- Follow the instructions below to complete the code for EACH section of QUESTION 1, as described in QUESTION 1.1. to QUESTION 1.4.

Example of the graphical user interface (GUI):



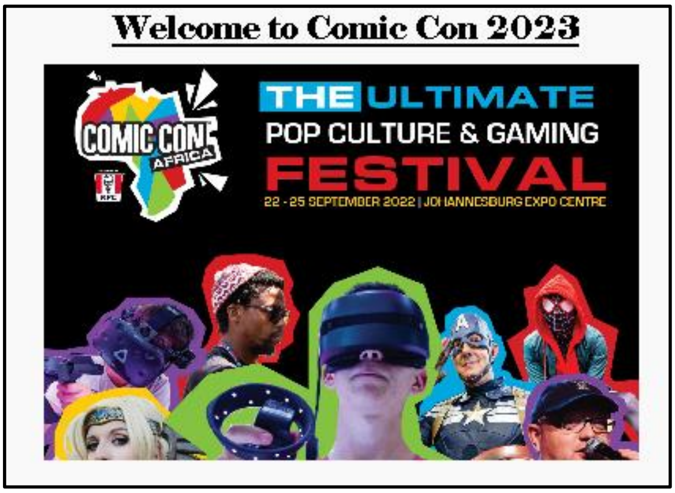
1.1 **Tab sheet [Welcome]: Button [Question 1.1]** (8)

Write code to do the following:

- Change the Image **imgPic** as follows:
  - Set the stretch property to true.
  - Set the width of the image to 377.
- Change the Label **lblWelcome** as follows:
  - Display the heading "Welcome to Comic Con 2023".

- ○ Underline the heading.
- ○ Change the font to 'Elephant'.

Example of output:



Welcome to Comic Con 2023

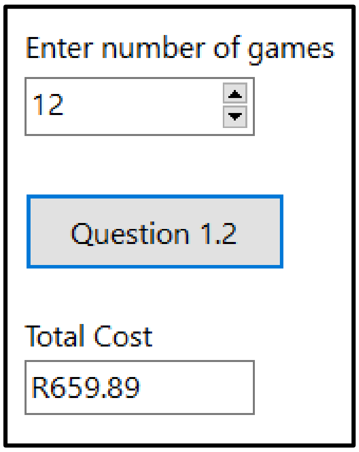
1.2 **Tab sheet [Game Cost]: Button [Question 1.2]** (5)

Educational games are expensive but when users buy games in bulk they can qualify for a discount.

Write code to do the following:

- Store input from the user for the number of games they would like to purchase in the provided variable **iGames**.
- Determine the cost of the games as follows and store it in the variable provided **rCost**:
  - ○ For every 10 games purchased the user gets one game for free.
  - ○ Use the constant variable provided, holding the cost per game, to determine the final cost.
- Display the final cost in **edtCost**.

Example of output when 12 games were entered:



Enter number of games

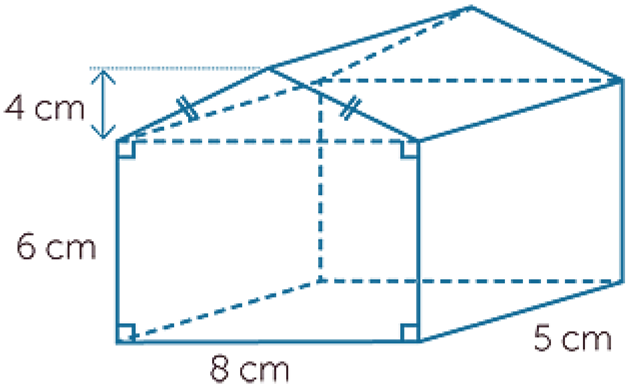
12

Total Cost

R659.89

1.3 Tab sheet [Volume]: Button [Question 1.3] (10)

A mathematical game needs to determine the volume of a model house consisting of a triangular prism as the roof and a rectangular prism as the base of the house.

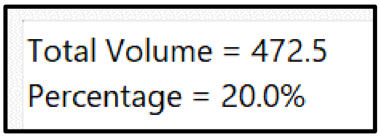


Declarations of the variables needed have been provided. Code has been provided to assign a value of 6 to the height of the house.

Write code to do the following:

- Make use of dialog boxes to store the input from the user for the length and width of the base of the model house and the height of just the roof.
- Determine the volume of the roof and the entire house making use of the following formulas:
    - Volume of triangular prism:
      1 / 2 X Roof Height X Width X Length
    - Volume of rectangular prism:
      Length X Width X House Height
- Calculate and display the percentage that the volume of the roof is of the total house volume.
- Display the volume of the house and the roof percentage, both formatted to ONE decimal place in the rich edit **redOutput**.

Example of output when 3 was entered as the height of the roof and 7 and 9 were entered for the width and length of the house respectively.



```
Total Volume = 472.5
Percentage = 20.0%
```

1.4 Tab sheet [Collatz]: Button [Question 1.4] (17)

The Collatz conjecture states that if you take any positive integer and repeatedly apply the following steps to find the next number;

- if the number is even, divide it by 2,
- if the number is odd, multiply it by 3 and add 1
- the sequence of numbers will always reach 1.

A number for which this is not true has never been found, but the conjecture has never been mathematically proven to be true. A program would make it easier to test the conjecture on multiple values.

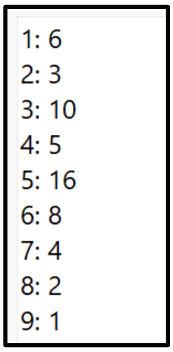
Write code to do the following:

- The user must input a number. Validate that this input is an integer number. If the input is invalid, write code to:
  - Clear the edit.
  - Place the cursor in the edit.
  - Change the colour of the edit to yellow.
  - Do not allow the rest of the code to execute.

- Continue to apply the calculations described below to determine the sequence until the number equals one. The number referred to below would first be the original number entered and thereafter it would be the answer to the previous calculation:
  - If the number is even, divide it by 2 to determine the next number.
  - If the number is odd then multiply it by 3 and add 1 to determine the next number.

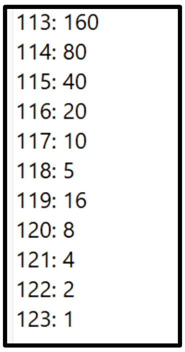
- Count how many numbers are generated before a value of one is reached.
- Display each new term and the count in the rich edit **redCollatz**.

Example of output when 6 was entered:



```
1: 6
2: 3
3: 10
4: 5
5: 16
6: 8
7: 4
8: 2
9: 1
```

An extract of output when 259 was entered:



```
113: 160
114: 80
115: 40
116: 20
117: 10
118: 5
119: 16
120: 8
121: 4
122: 2
123: 1
```

- Enter your name and surname as a comment in the first line of the program file.
- Save all the files ('File/Save All').
- Print the code if required.

TOTAL SECTION A [40]

SECTION B

QUESTION 2 : SQL AND DATABASE PROGRAMMING

Various Companies are selling super hero action figures at the Comic Con. A database has been set up by the organisers to keep track of the companies and which action figures they sell.

An application has been created that will use the **Heroes.mdb** database.

The database contains two tables called **tblCompany** and **tblFigure.**

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

Do the following:

- Open the incomplete project file called **Question2_p.dpr** in the **Question 2** folder.

- Enter your name and surname as a comment in the first line of the **Question2_u.pas** unit 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**.
- 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.

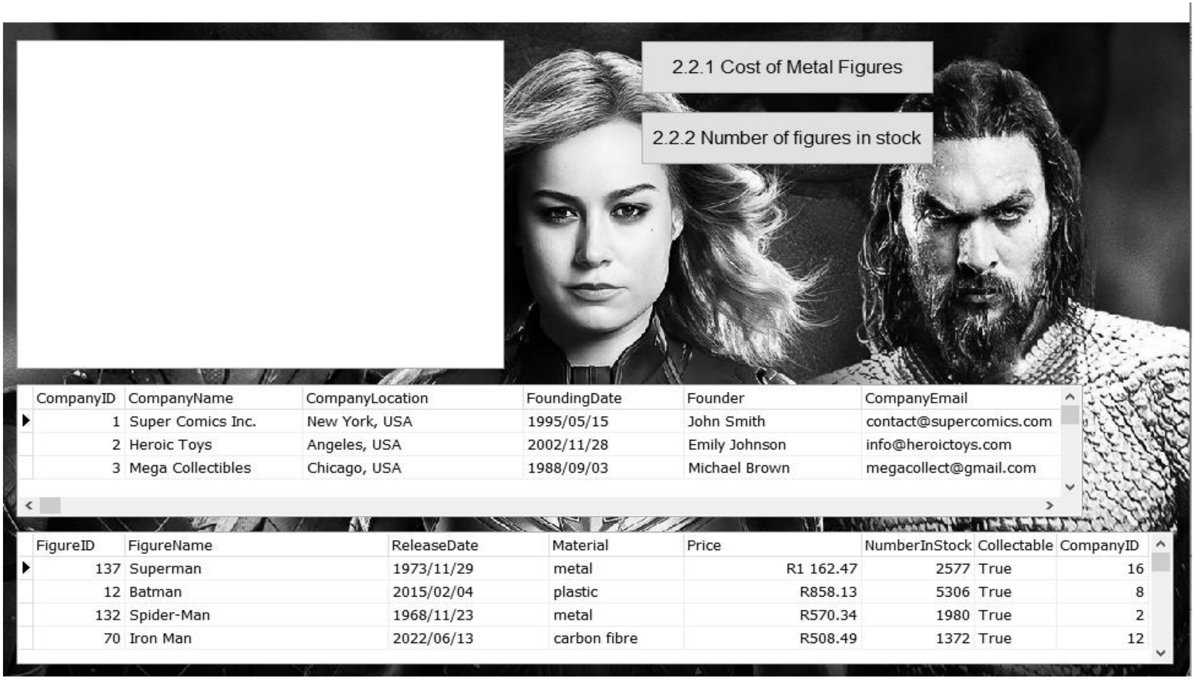
NOTE:

- 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.
- TWO variables are declared as public variables, as described in the table below.

Variable	**Data type**	**Description**
tblCompany	TADOTable	Refers to the table **tblCompany**
tblFigure	TADOTable	Refers to the table **tblFigure**

2.1 Tab sheet [Question 2.1]

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



CompanyID	CompanyName	CompanyLocation	FoundingDate	Founder	CompanyEmail
1	Super Comics Inc.	New York, USA	1995/05/15	John Smith	contact@supercomics.com
2	Heroic Toys	Angeles, USA	2002/11/28	Emily Johnson	info@heroictoys.com
3	Mega Collectibles	Chicago, USA	1988/09/03	Michael Brown	megacollect@gmail.com

FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
137	Superman	1973/11/29	metal	R1 162.47	2577	True	16
12	Batman	2015/02/04	plastic	R858.13	5306	True	8
132	Spider-Man	1968/11/23	metal	R570.34	1980	True	2
70	Iron Man	2022/06/13	carbon fibre	R508.49	1372	True	12

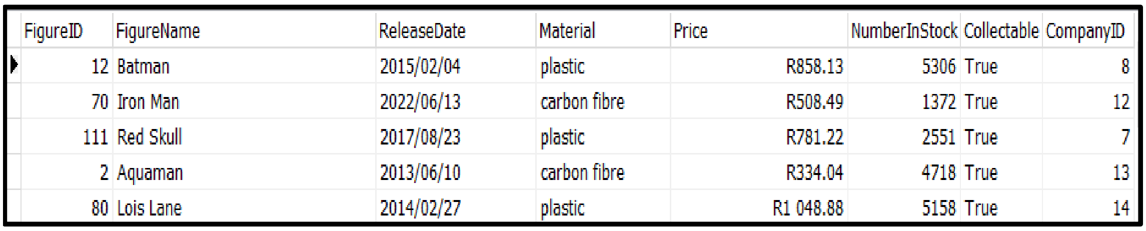
NOTE:

- Use ONLY SQL code to answer QUESTION 2.1.1 to QUESTION 2.1.5.
- Code to execute the SQL statements and display the results of the queries is provided. The SQL statements that will be assigned to the string **sSQL** are incomplete.
- Complete the SQL statements to perform the tasks described in QUESTION 2.1.1 to QUESTION 2.1.5 below.

2.1.1 Button [2.1.1 – Collectable Action Figures] (5)

Display all details of action figures which are collectable and that had a release date anytime in the 21st century i.e. after the 31st of December 1999.

Example of output:

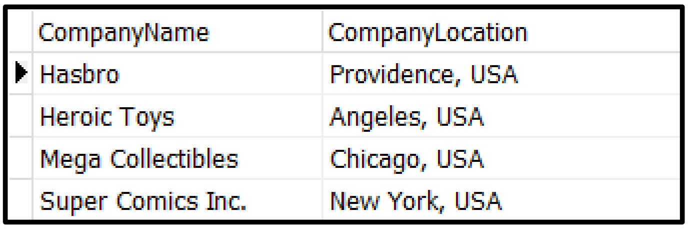


FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
12	Batman	2015/02/04	plastic	R858.13	5306	True	8
70	Iron Man	2022/06/13	carbon fibre	R508.49	1372	True	12
111	Red Skull	2017/08/23	plastic	R781.22	2551	True	7
2	Aquaman	2013/06/10	carbon fibre	R334.04	4718	True	13
80	Lois Lane	2014/02/27	plastic	R1 048.88	5158	True	14

2.1.2 **Button [2.1.2 – Companies in the USA]** (4)

Display the company name and company location of all companies that are located in the USA.

Example of output:



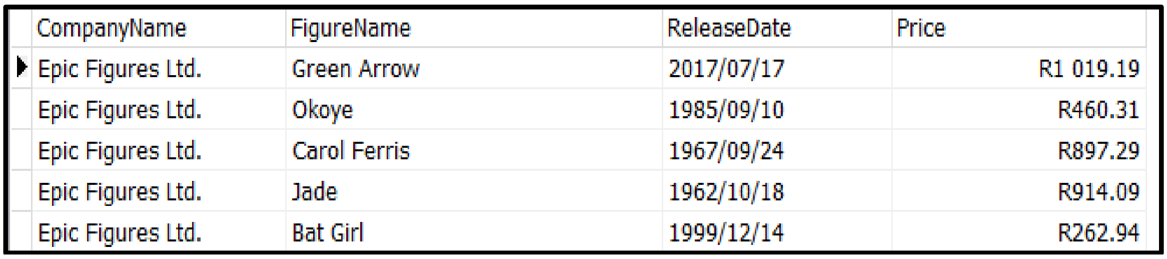
	CompanyName	CompanyLocation
▶	Hasbro	Providence, USA
	Heroic Toys	Angeles, USA
	Mega Collectibles	Chicago, USA
	Super Comics Inc.	New York, USA

2.1.3 **Button [2.1.3 – All Toys by Company]** (4)

Display company name, the figure name, the release date, and the price of all toys that have a company name matching the name selected in the combo box **cmbCompany**.

Code has been provided to extract the company name from the combo box and store it in a variable named **sSearch**.

Example of output if Epic Figures Ltd. was chosen:

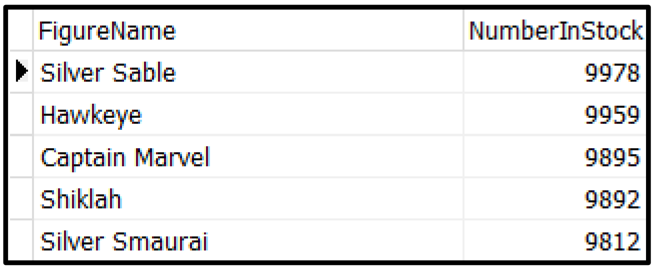


	CompanyName	FigureName	ReleaseDate	Price
▶	Epic Figures Ltd.	Green Arrow	2017/07/17	R1 019.19
	Epic Figures Ltd.	Okoye	1985/09/10	R460.31
	Epic Figures Ltd.	Carol Ferris	1967/09/24	R897.29
	Epic Figures Ltd.	Jade	1962/10/18	R914.09
	Epic Figures Ltd.	Bat Girl	1999/12/14	R262.94

2.1.4 **Button [2.1.4 – Stock Numbers]** (6)

Display the figure name and the number in stock of all action figures whose number in stock is greater than the average number of action figures in stock.

Example of output:



	FigureName	NumberInStock
▶	Silver Sable	9978
	Hawkeye	9959
	Captain Marvel	9895
	Shiklah	9892
	Silver Smaurai	9812

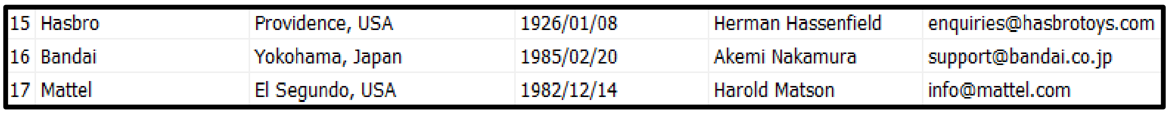
2.1.5 Button [2.1.5 – Add Company] (4)

Write a SQL statement to add a company to **tblCompany** using the details below:

Company ID:	17
Company Name:	Mattel
Company Location:	El Segundo, USA
Date Founded:	12/14.1982
Founder:	Harold Matson
Company email:	info@mattel.com

Code has been provided to display the database after the company has been added

Example of output:



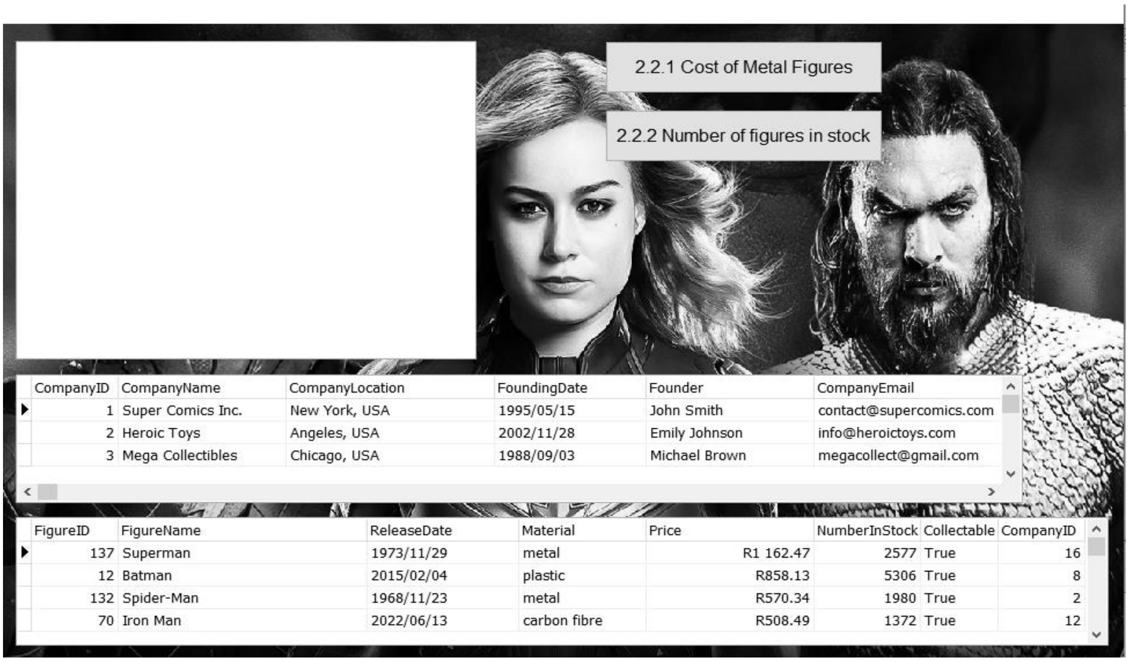
15	Hasbro	Providence, USA	1926/01/08	Herman Hassenfield	enquiries@hasbrotoys.com
16	Bandai	Yokohama, Japan	1985/02/20	Akemi Nakamura	support@bandai.co.jp
17	Mattel	El Segundo, USA	1982/12/14	Harold Matson	info@mattel.com

2.2 Tab sheet [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.

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



CompanyID	CompanyName	CompanyLocation	FoundingDate	Founder	CompanyEmail
1	Super Comics Inc.	New York, USA	1995/05/15	John Smith	contact@supercomics.com
2	Heroic Toys	Angeles, USA	2002/11/28	Emily Johnson	info@heroictoys.com
3	Mega Collectibles	Chicago, USA	1988/09/03	Michael Brown	megacollect@gmail.com

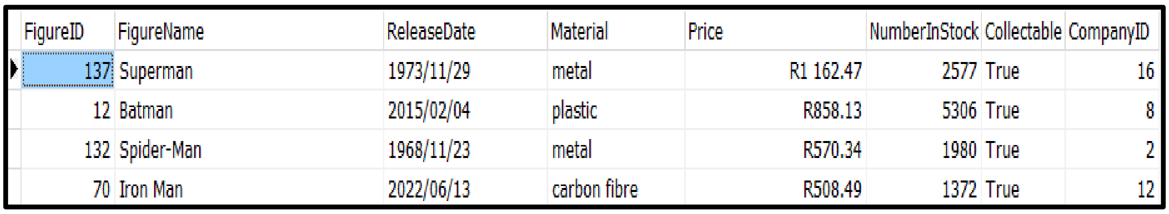
FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
137	Superman	1973/11/29	metal	R1 162.47	2577	True	16
12	Batman	2015/02/04	plastic	R858.13	5306	True	8
132	Spider-Man	1968/11/23	metal	R570.34	1980	True	2
70	Iron Man	2022/06/13	carbon fibre	R508.49	1372	True	12

2.2.1 [Button – Cost of Metal Figures] (5)

Recently, there has been a sharp increase in the cost of metal which is required in the manufacture of certain action figures.

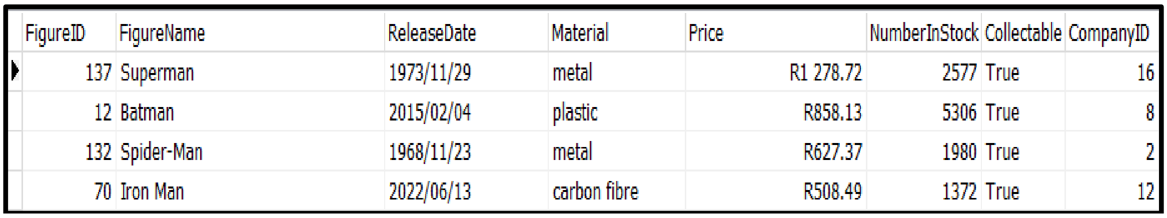
Write code to increase the price of all action figures manufactured with metal by ten percent.

Example of output before btnQ2_2_1 was clicked:



FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
137	Superman	1973/11/29	metal	R1 162.47	2577	True	16
12	Batman	2015/02/04	plastic	R858.13	5306	True	8
132	Spider-Man	1968/11/23	metal	R570.34	1980	True	2
70	Iron Man	2022/06/13	carbon fibre	R508.49	1372	True	12

Example of output after btnQ2_2_1 was clicked:

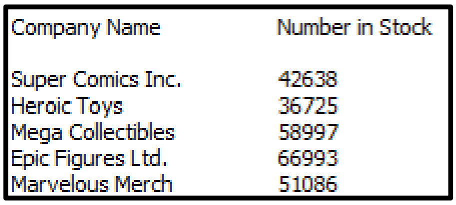


FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
137	Superman	1973/11/29	metal	R1 278.72	2577	True	16
12	Batman	2015/02/04	plastic	R858.13	5306	True	8
132	Spider-Man	1968/11/23	metal	R627.37	1980	True	2
70	Iron Man	2022/06/13	carbon fibre	R508.49	1372	True	12

2.2.2 [Button – Number of figures in stock] (12)

Write code to determine the total number of action figures that each company has in stock currently. Display the results in the rich edit **redQ2**.

Example output when btnQ2_2_2 was clicked:



```
Company Name             Number in Stock

Super Comics Inc.        42638
Heroic Toys              36725
Mega Collectibles        58997
Epic Figures Ltd.        66993
Marvelous Merch          51086
```

- Enter your name and surname as a comment in the first line of the program file.
- Save all the files ('File/Save All').
- Print the code if required.

TOTAL SECTION B: [40]

SECTION C

QUESTION 3: OBJECT ORIENTATED PROGRAMMING

New exhibitors at Comic Con need to be registered and the cost for the area (stand) that they are renting needs to be calculated. The cost for the stand includes walling, a name board, carpeting, spotlights, plug points, a 50 word entry into the online exhibitor's directory and badges for the exhibitors to wear (based on the size of the stand).

Do the following:

- Open the incomplete program in the **Question 3** folder
- Open the incomplete object class **TRegisteration_u**

The class already contains a **toString** method to display the values of the attributes.

Complete the rest of the code as described in QUESTION 3.1 and QUESTION 3.2 to add functionality to the program.

3.1 The incomplete code in the registration class (**TRegistration**) contains the declaration of the five attributes which describe the elements in the registration class.

The attributes for a Registration object have been declared as follows

Attribute	Description
fName	Name of the exhibitor
fStandSize	Size of the stand required in square meters
fCost	Cost for renting the stand
fProfile	A 50 word profile of the exhibitor for the directory entry.
fBadges	The number of badges allocated to the exhibitor.

Complete the code in the object class as described in QUEST 3.1.1 to QUEST 3.1.6 below :

3.1.1 Write code to create a **constructor** method which receives the following parameter values: the name of the exhibitor and the size of the stand required in square meters. Assign the relevant attributes to their respective parameter values. Set the *fCost and fBadges* attributes to a value of 0 and the *fProfile* attribute to an empty string. (4)

3.1.2 Write a mutator method, **calcCost,** which calculates the amount due for the rental of the stand for the duration of Comic Con (4 days) by using the size of the stand required. This method must change the *fcost* attribute. (6)

The cost for the stand **for one day** is calculated as shown in the table below.

Stand Space m^2	**Cost per m^2**
1 - 18	R3690 per m^2
19 - 36	R3585 per m^2
>= 37	R3447 per m^2

(2)

3.1.3 Write code for a mutator method, **setProfile** which has a string parameter containing the profile description. Use this value to change *fProfile.*

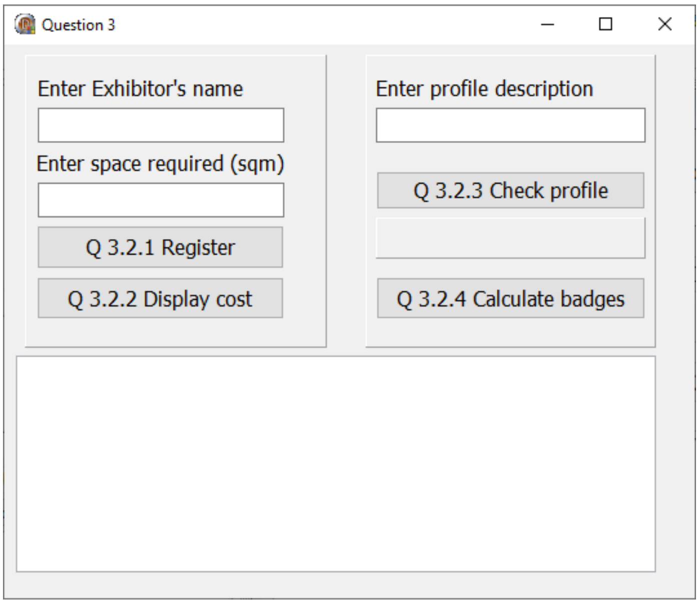
3.1.4 Write code for a method, **checkWords** which checks the attribute fProfile and returns a Boolean value of **true** if the number of words in the profile is less than or equal to 50 and **false** if it exceeds 50. (8)

3.1.5 Write code for a mutator method **calcBadges** which calculates the number of badges allocated to the exhibitor. This method must change the variable *fBadges*. (3)

An exhibitor is entitled to a badge for every 9m^2 of space rented rounded up to the nearest whole.

3.1.6 Write code for a method **getBadges** which retrieves the number of badges the exhibitor is eligible from the private attribute. (2)

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 **objRegistration.**



Write code to perform the tasks described in QUEST 3.2.1 to QUEST 3.2.4

3.2.1 Button [3.2.1 Register] (4)

Write code to do the following:

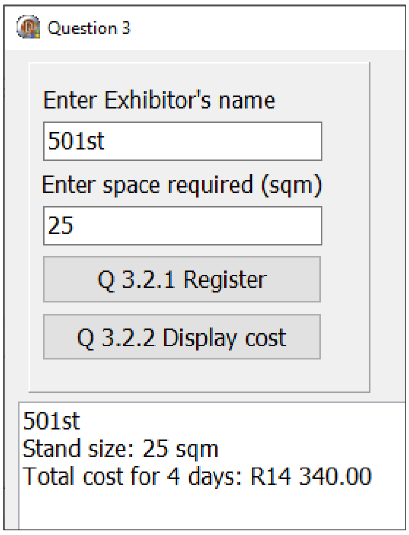
- Read the name of the exhibitor and the stand space required from the respective components.
- Using this information, instantiate a new Registration object.

3.2.2 Button [3.2.2 Display cost] (4)

Write code to do the following :

- Call the **appropriate** method to calculate the cost for the rental of the stand.
- Use the **toString** method to display the exhibitor's details in **redQ3**.

Example of output



Enter Exhibitor's name: 501st

Enter space required (sqm): 25

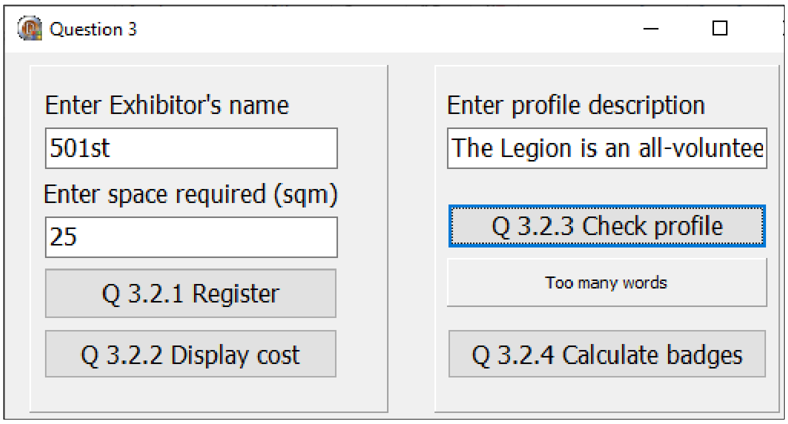
```
501st
Stand size: 25 sqm
Total cost for 4 days: R14 340.00
```

3.2.3 Button [3.2.3 Word limit] (4)

Write code to do the following:

- Retrieve the profile description from the edit.
- Call up the parametrized method to update the profile in the class.
- Call up the method to check if the profile description does not exceed the allowed number of words.
- If the profile description is the correct length, display the message **"Profile approved for the directory"** in **pnlQ3_2_3.**
- Otherwise display the message **"Too many words"** in the panel**.**
- To save time, a textfile, **Exhibitor.txt** has been created with profile descriptions that you can copy and paste into the edit to test your program. No marks will be given for accessing this textfile with code.

Example of output for exhibitor 501st (next page)



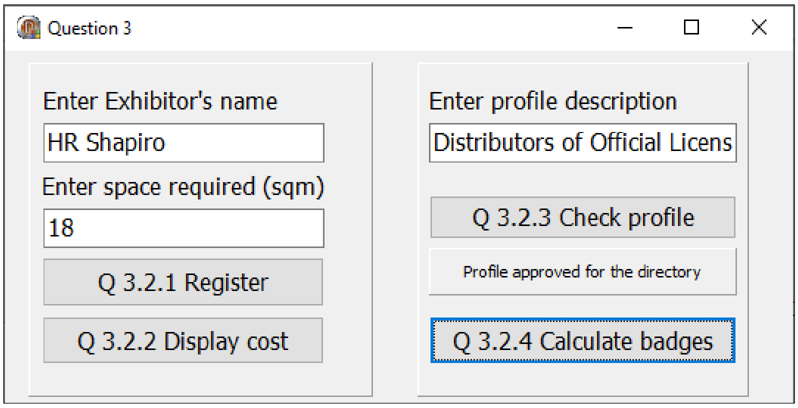
501st

25

The Legion is an all-voluntee

Too many words

Example of output for exhibitor HR Shapiro



HR Shapiro

18

Distributors of Official Licens

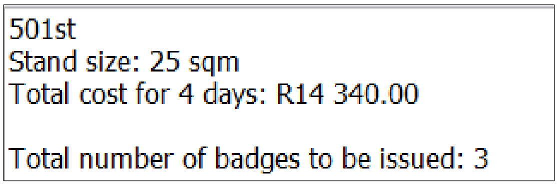
Profile approved for the directory

3.2.4 **Button [3.2.4 Badges]** (3)

Write code to do the following:

- Call the **appropriate methods** to calculate the number of badges that the exhibitor is eligible for and display the number of badges in **redQ3**.

Example of output for a stand size of 25 m^2



```
501st
Stand size: 25 sqm
Total cost for 4 days: R14 340.00

Total number of badges to be issued: 3
```

- Enter your name and surname as a comment in the first line of the program file.
- Save all the files ('File/Save All').
- Print the code if required.

TOTAL SECTION C: [40]

SECTION D

QUESTION 4: PROBLEM-SOLVING PROGRAMMING

SCENARIO

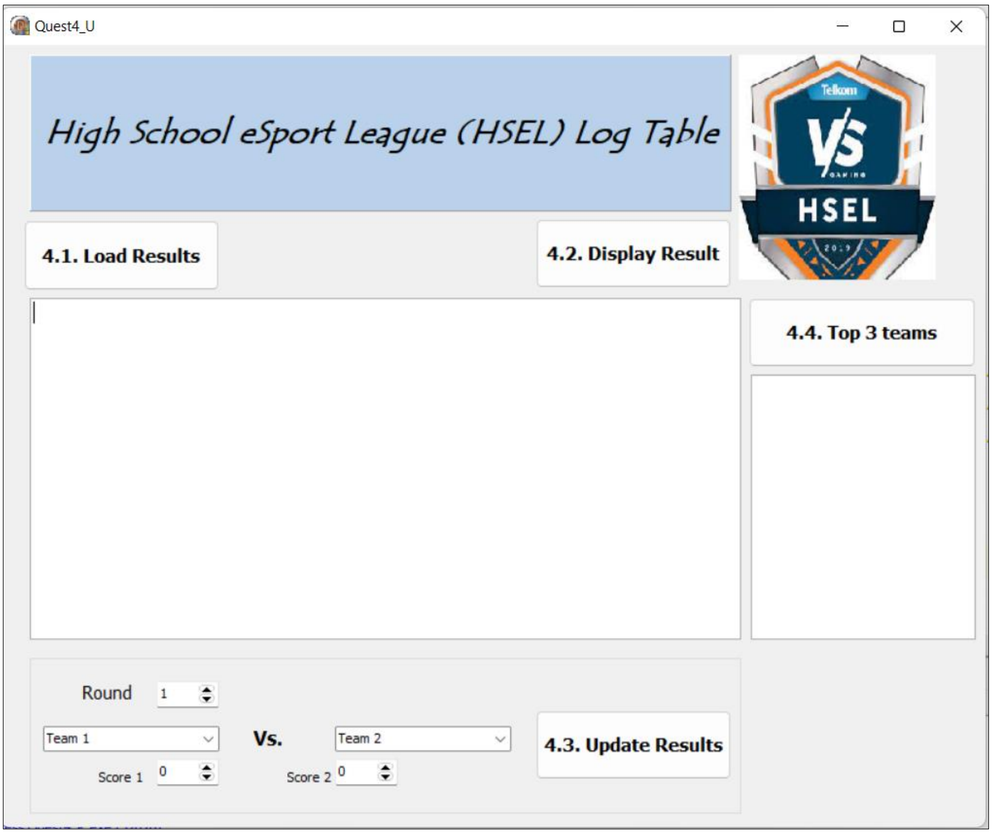
HSEL (High School eSport League) in collaboration with TelkomSA organize eSport matches between schools to qualify to Comic Cons. Eight school teams compete in five rounds in a round robin format and the top three teams in the league qualify for the finals at Comic Cons. You are required to design a program to manage results of the qualify round and determine the top 3 teams that qualify to the Comic Cons.

Do the following:

Compile and execute the program in the **Quest4** folder. Currently the program has no functionality.

Complete the code for each question, as described in QUESTION 4.1 to QUESTION 4.4.

Supplied GUI:



Supplied data:

You are provided with a two-dimensional array called **arrResult** that is populated with the default character (*), at form activate event, which represents no result is captured for a given fixture. Alternatively, you are provided with **arrResult2** loaded with identical data as the data in text file **Results.txt** should you fail to write code to load the data from text file to the 2D array **arrResult** as per the specifications in question 4.1. (Note that you will be awarded 0/7 marks for not meeting the requirements in question 4.1.)

You are also provided with a text file called **Results.txt** that stores the number of wins a team has (best of 3 games) formatted as:

Round#Team1#Team2#Team1Wins#Team2Wins

Eg. 1#NWD#OIS#2#1
Round = any round between 1 to 5 in the fixture list. (Round 1)
Team1 = the first team in the fixture (NWD-Northwood School team)
Team2 = the second team in the fixture (Orient Islamic School team)
Team1Wins = number of wins for team1 out of 3games (NWD won 2 out of 3 matches)
Team2Wins = score of 1 to 3 for the second team (OIS won 1 of 3 matches)

Output area:

Rich edit **redDisplay** is provided to display the content of the two-dimensional array **arrResult**. As well as rich edit **redQ4_4** to display the top three teams in the league log table.

NOTE:

- You are NOT allowed to modify supplied data manually. Code must be written to manipulate the supplied data according to the requirements.

- Good programming techniques and modular design must be applied in the design and coding of your solution.

- **NO marks will be assigned for hardcoding**. Use control structures and variables where necessary.

- The following 8 schools are participating in the HSEL league with 5 round of fixtures
  - NWD – Northwood School
  - OIS – Orient Islamic School
  - DHS – Durban Boys High School
  - GCB – George Campbell
  - DGH – Danville Girls High School
  - NGH – Northland Girls High
  - WVB – Westville Boys High
  - EDC – Eden College

4.1 Button [Load Results]

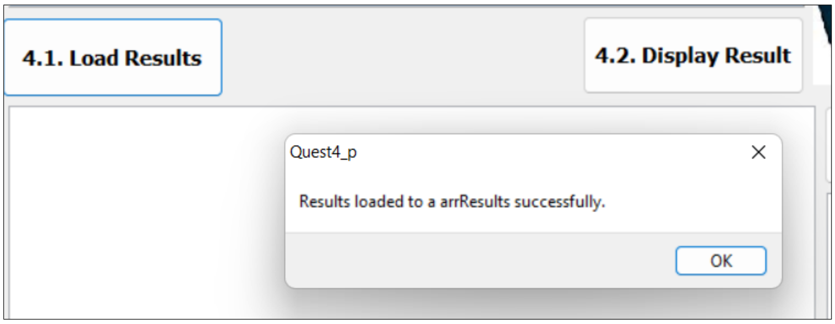
The program must transfer/store each result from the text file **Results.txt** to the corresponding elements in the 2D array **arrResults**.

Write code to do the following:

- Open the text file **Results.txt** for reading.
- Read line from the text file and split the contents using the # delimiter to determine the round number, two team names and their corresponding scores in the fixture.
- The team with more number of wins will be awarded 3 points, otherwise 0 points.
  Eg. 1#DHS#WVB#2#1
  DHS will be awarded 3 points at array position arrResults[3,1]
  WVB will be awarded 0 points at array position arrResults[7,1]

  Refer for sample output in **question 4.2**.

- Display message to indicate that data is loaded from text file **Result.txt** to **arrResult**.

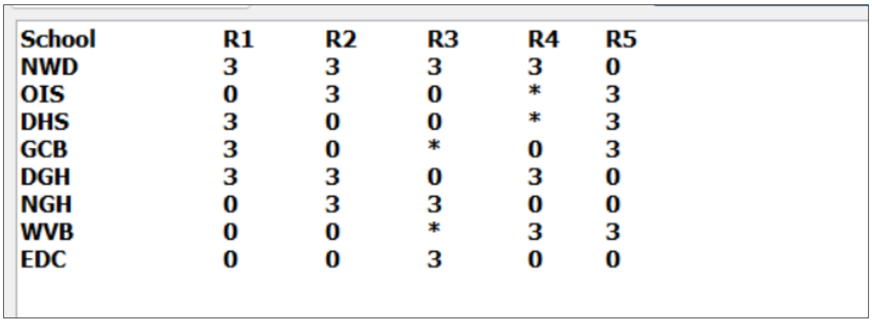


Results loaded to a arrResults successfully.

NOTE: If you fail to 'Load Results' from the text file **Results.txt** to **arrResult**, you are provided **arrResult2** as a comment. You may uncomment and use arrResult2 to continue to work with question 4.2 – 4.4 on a condition that **NO MARK** will be awarded to you for 4.1. i.e. You will get 0/7 for question 4.1. (7)

4.2. Button [Display Results]

You are required to display the content of **arrResults** as shown below, note that the (*) in the display indicates that results for the corresponding fixtures are not captured. Eg. result for the match GCB vs. WVB in round 3 is not recorded in the text file, hence the array element at the corresponding position remains (*) as default.



```
School          R1      R2      R3      R4      R5
NWD             3       3       3       3       0
OIS             0       3       0       *       3
DHS             3       0       0       *       3
GCB             3       0       *       0       3
DGH             3       3       0       3       0
NGH             0       3       3       0       0
WVB             0       0       *       3       3
EDC             0       0       3       0       0
```

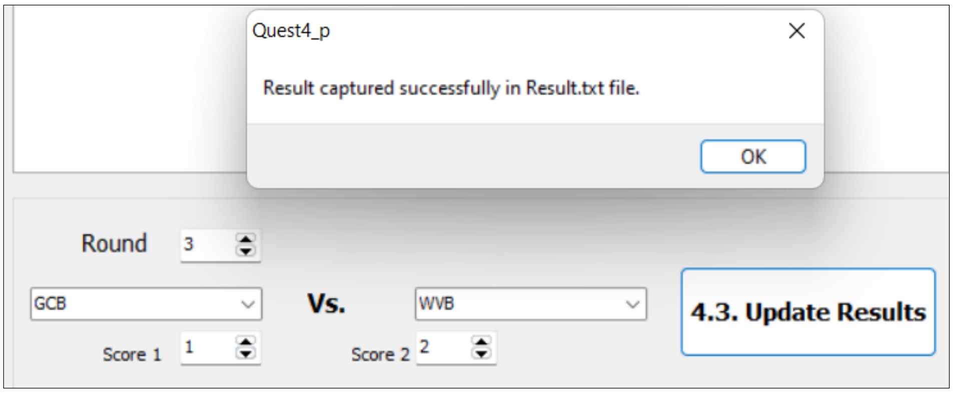
(5)

4.3 Button [Update Result]

Due to technical issues, there were two fixtures that were rescheduled to take place on another day. Results are now ready to be captured and need to be added to the text file **Results.txt**. Write code to write the results of the outstanding fixtures from the GUI components provided for question 4.3 to **Results.txt**. Users are required to capture the following two outstanding results one at a time.

SAMPLE OUTPUT:

Result 1 : 3#GCB#WVB#1#2



Quest4_p

Result captured successfully in Result.txt file.

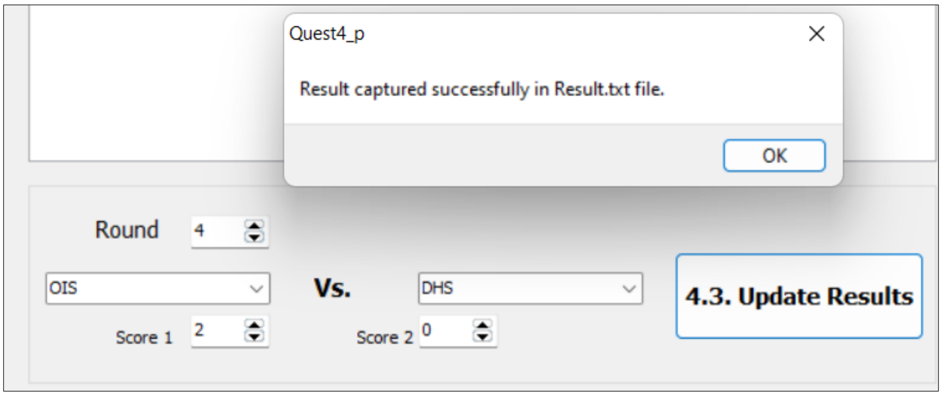
Round 3

GCB Vs. WVB

Score 1 1

Score 2 2

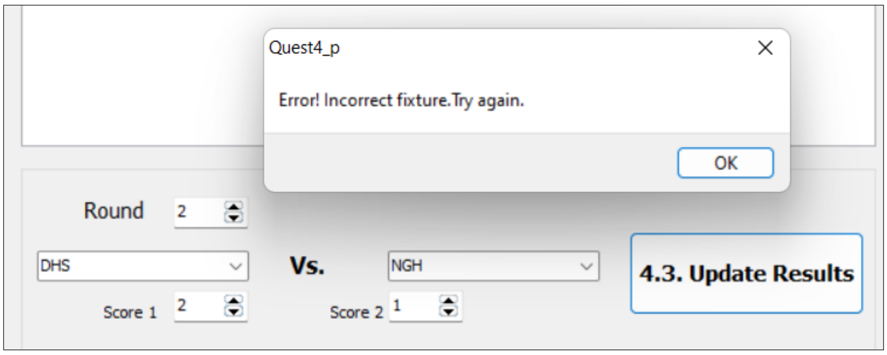
Result 2 : 4#OIS#DHS#2#0



Quest4_p

Result captured successfully in Result.txt file.

If a user attempts to capture a result which is already uploaded, your program should indicate that the result is not valid.



Quest4_p

Error! Incorrect fixture.Try again.

(8)

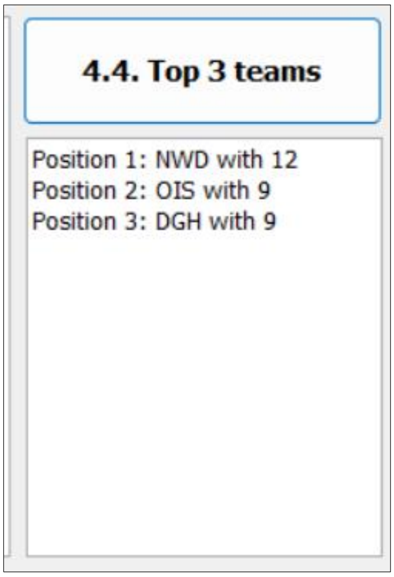
4.4 Button [Top 3 teams]

Write a program to calculate the total point each team accumulated from the 5 rounds of fixtures and determine the top 3 teams that qualify for the finals in Comic Cons.

Do the following:

- Create 1D array called **arrTotPoints**.
- Calculate totals for each row that represents total points from the 5 rounds for a the corresponding team in **arrParticipants.**
- Sort the array arrTotPoints from highest to lowest and their corresponding team names from arrParticipants.
- Display the first three elements in the sorted array as the sample output below.

Example of output:



```
Position 1: NWD with 12
Position 2: OIS with 9
Position 3: DGH with 9
```

(10)

TOTAL SECTION D: [30]

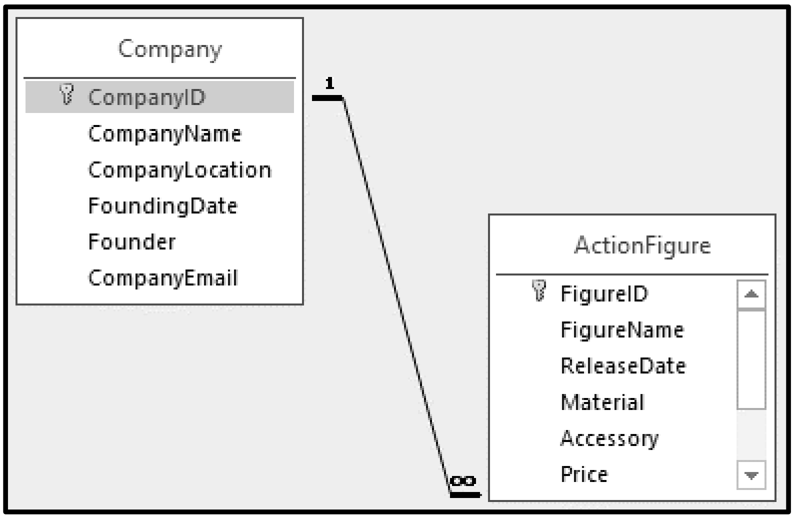
GRAND TOTAL = 150

INFORMATION TECHNOLOGY P1

DATABASE INFORMATION QUESTION 2:

The database **HeroesDB** consists of table **tblCompany** and **tblFigure**.

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



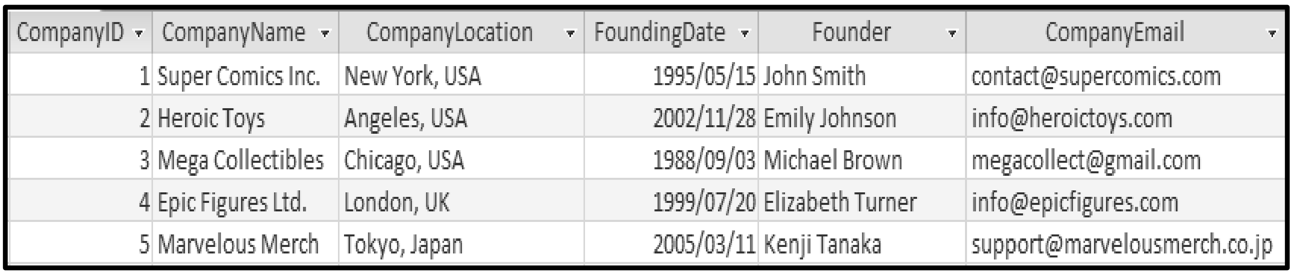
The design of the database tables is as follows:

Table: **tblCompany**

This table contains details of the clients.

Field name	**Data type**	**Description**
CompanyID	Number	Integer value that is the unique number for each company
CompanyName	Text (20)	The name of the company
CompanyLocation	Text (25)	The location of the company
FoundingDate	Date/Time	The date on which the company was founded
Founder	Text (20)	The name and Surname of the company founder
CompanyEmail	Text (40)	The email address of the company

Example of the records in the tblCompany table:



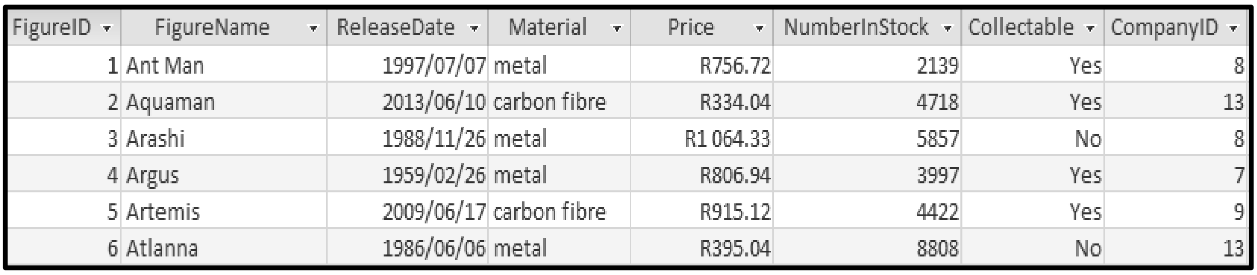
CompanyID	CompanyName	CompanyLocation	FoundingDate	Founder	CompanyEmail
1	Super Comics Inc.	New York, USA	1995/05/15	John Smith	contact@supercomics.com
2	Heroic Toys	Angeles, USA	2002/11/28	Emily Johnson	info@heroictoys.com
3	Mega Collectibles	Chicago, USA	1988/09/03	Michael Brown	megacollect@gmail.com
4	Epic Figures Ltd.	London, UK	1999/07/20	Elizabeth Turner	info@epicfigures.com
5	Marvelous Merch	Tokyo, Japan	2005/03/11	Kenji Tanaka	support@marvelousmerch.co.jp

Table: **tblFigure**

This table contains information of all the action figures.

Field name	**Data type**	**Description**
FigureID	Number	Unique code for each action figure
FigureName	Text (30)	Date of the collection
ReleaseDate	Date/Time	The date on which the action figure was first released
Material	Text (15)	The material that the figure is made from
Price	Currency	The selling price of the action figure
NumberInStock	Number	An integer number that records the number of action figures of each type the company has in stock.
Collectable	Yes/No	A Boolean value that determines whether the action figure is part of a collection or not.
CompanyID	Number	Integer value that is the unique number for each company

Example of the records in the tblFigure table:



FigureID	FigureName	ReleaseDate	Material	Price	NumberInStock	Collectable	CompanyID
1	Ant Man	1997/07/07	metal	R756.72	2139	Yes	8
2	Aquaman	2013/06/10	carbon fibre	R334.04	4718	Yes	13
3	Arashi	1988/11/26	metal	R1 064.33	5857	No	8
4	Argus	1959/02/26	metal	R806.94	3997	Yes	7
5	Artemis	2009/06/17	carbon fibre	R915.12	4422	Yes	9
6	Atlanna	1986/06/06	metal	R395.04	8808	No	13

NOTE:

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