



Province of the
EASTERN CAPE
EDUCATION

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2020



INFORMATION TECHNOLOGY P1

MARKS: 150

TIME: 3 hours

This question paper consists of 21 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into FOUR sections. Candidates must answer ALL the questions in 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 question paper is set with programming terms that are specific to the Delphi programming language.
4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.
5. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.
6. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.
7. Routines, such as locate, search, sort and selection, must be developed from first principles. You may NOT use the built-in features of Delphi for any of these routines.
8. All data structures must be declared by you, the programmer, unless the data structures are supplied.
9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.
10. Make sure that your name and surname appear 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. The files that you need to complete this question paper have been given to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **20%tRiAL#gR12**

Once extracted, the following list of files will be available in the folder **DataSept2020**:

Question 1:

Question1_u.pas
Question1_u.dfm
Question1_p.dpr
Question1_p.res

Question 2:

Trees.jpg
Question2ClassDefinition.pas
Question2_u.pas
Question2_u.dfm
Question2_p.dpr
Question2_p.res



Question 3:

dbConnection_u.pas
WasteDisposal.mdb
WasteDisposalBackup.mdb
Question3_u.pas
Question3_u.dfm
Question3_p.dpr
Question3_p.res

Question 4:

waste.txt
Question4_u.pas
Question4_u.dfm
Question4_p.dpr
Question4_p.res

QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

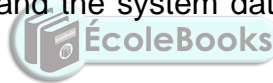
- Open the incomplete program in the **Question 1** folder.
- Enter your full name as a comment in the first line of the **Question1_u.pas** file.
- Compile and execute the program. The program has no functionality currently.
- Follow the instructions below to complete the code for each section of QUESTION 1, as described in QUESTION 1.1, QUESTION 1.2, QUESTION 1.3 and QUESTION 1.4.

Your school has created a club to provide information and conduct research on various environmental issues. Complete the program for the club by using the instructions below.

1.1 Button [1.1 Display Club name]

Write code in the Q1.1 event handler to do the following:

- Obtain the name of the club from the edit box named **edtclub**.
- The system date must be displayed at the end of the name of the club.
- The font of the panel must be set to size 24 and the style of the font must be bold.
- The name of the club and the system date must be displayed in the panel named **pnlclub**.



Example of output:

(6)

1.2 Button [1.2 Process]

Your school has collected paper, plastic and glass for recycling. The program in the Question 1 folder contains components to keep track of how much the monetary value is for each type of waste material that has been collected. A user will choose a waste material type and then enter the weight for that product. The refund value in Rands for each waste material type must be added up and displayed on the panels provided for paper, plastic and glass.

Write code to do the following:

- Obtain the input from the edit box named **edtweight**. This number is entered as **grams**.
- Use the choice selected from the radiogroup box named **rgpchoice**.
- The unit price paid for each type of waste material is listed below:
 - Paper = R25 per **kilogram**
 - Plastic = R35 per **kilogram**
 - Glass = R40 per **kilogram**

The total price paid for each waste material must be calculated as follows:

- Convert the weight in grams to kg by dividing by 1000.
- Only whole kilograms will be considered for payment and any amount less than 1 kg will not be included. Do NOT round the amount after dividing by 1000.
- Multiply the weight in kg by the correct unit price paid for a waste material type in order to obtain the amount to be paid.
- Add the amount to be paid to a total to be paid for that waste material type.
- Display the total paid for each waste material on the correct panels provided (**pnlpaper**, **pnlplastic** and **pnlglass**).

Use the following data to test your solution:

Material	Weight in g	Weight in kg	Monetary value
Paper	356	0	R 0
Paper	1785	1	R 25
Paper	2356	2	R 50
Plastic	3500	3	R 105
Plastic	7854	7	R 245
Glass	1000	1	R 40
Glass	3745	3	R 120

Example of output after entering the data in the table above:

(16)

1.3 Button [1.3 Test for leap year]

A leap year has 366 days instead of 365 days. Write code in the onclick event of Q1.3 button to test if a year entered is a leap year.

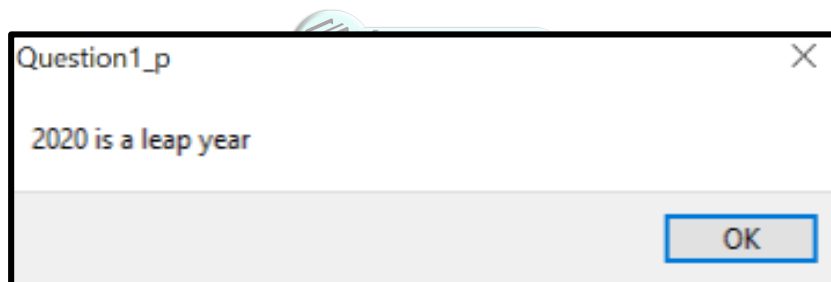
To check whether a year is a leap year or not, you need to check the following 3 conditions:

1. Any year that is divisible by 400 with no remainder is definitely a leap year.
2. If it is not divisible by 400, then check if it is divisible by 100. If it is divisible by 100 with no remainder, then it is NOT a leap year (even if it is divisible by 4 with no remainder).
3. If the above two conditions are not satisfied, then check for divisibility by 4. If it is divisible by 4 with no remainder then it is a leap year.

Do the following:

- Obtain the input from the edit box named **edtyear**.
- Use the algorithm given above to determine if the year that is entered is a leap year or not.
- Display a suitable message in a message component.

Example of output:



Data to test your solution:

Leap years from 2004 to 2044:

2004
2008
2012
2016
2020
2024
2028
2032
2036
2040
2044

(12)

1.4 Button [1.4 Lucky letters]

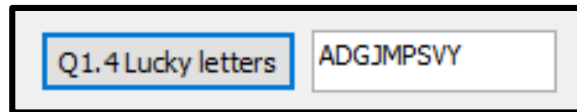
Lucky prizes will be awarded to members of the club at a meeting.

Every third letter of the alphabet will be the first letter of the surnames of the winners, starting with the letter 'A'.

Write code in the onclick event of Q1.4 button to display 9 letters of the alphabet, using the rule described above, in the richedit named **reddisplay**.

NOTE: String handling and iteration (loops) must be used.

Example of output:



(6)

- 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]



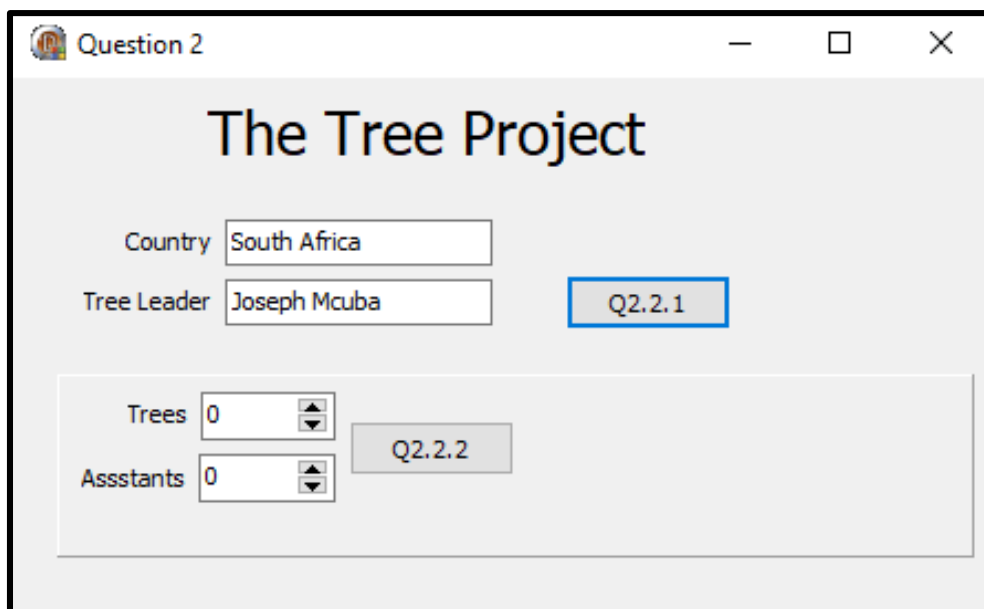
QUESTION 2: OBJECT-ORIENTED PROGRAMMING

A global organisation named 'The Tree Project' has branches around the world. A program is supplied in the QUESTION 2 folder to document the number of trees and workers as well as to calculate the monetary funds required to run this project in a particular country.

Do the following:

- Open the incomplete program in the Question 2 folder.
- Open the incomplete object class **Question2ClassDefinition.pas**.
- Enter your name as a comment in both **Question2ClassDefinition.pas** and **Question2_u.pas**.
- Compile and execute the program. Currently the program has no functionality.
- Do NOT remove or change any provided code.

The following user interface is displayed:



Complete the code for this program, as specified in QUESTION 2.1 and QUESTION 2.2.

- 2.1 The incomplete class (**Tcountry**) contains the declaration of five attributes that describe the **objcountry** object.

NAMES OF ATTRIBUTES	DESCRIPTION
fcountry	The name of the country
fleader	The name of the leader for the tree project in a country
ftrees	The number of trees to be planted
ffunds	The money required to run the project
fassistants	The number of people working on the project

Complete the code in the object class, as described in QUESTION 2.1.1 to QUESTION 2.1.4 below.

- 2.1.1 Write code for a **constructor** method named **Create** that will receive the name of a country and the name of a leader as parameters.

Assign these parameter values to the correct attributes and initialise the attributes for the number of trees and assistants to zero. (5)

- 2.1.2 Write code for a method named **calculatefunds** which must use the attributes named **ftrees** and **fassistants** to calculate and return the amount of funding needed for the tree project.

The funds are calculated as follows:

- The leader must receive R250
- The cost of planting one tree is R2
- Each assistant must receive R100

Example:

6 assistants and 100 trees: Cost of project = R250 + R600 + R200 = R1050 (6)

- 2.1.3 Write code for a method named **addnumbers** that will receive two integer parameters which will each represent the number of trees and the number of assistants with which you will increase the attributes of the class.

Set the value of the attributes as described below:

- Use the attributes named **ftrees** and **fassistants**.
- Increase these attributes with the values of the corresponding parameters
- Use the method written previously named **calculatefunds** to set the attribute named **ffunds**. (7)

- 2.1.4 Write code for a method named **tostring** which must return a string.

The string must contain the attributes of the class joined together in the following format:

<country name>

<leader name> and <number of assistants > assistants

<number of trees> trees

Funds: <amount of funds required in Rands>

Example of output:

```
South Africa
Joseph Mcuba and 20 assistants
100 trees
Funds: R2 450,00
```

(8)

2.2 An incomplete unit **Question2_u.pas** has been provided.

It contains code for the object class to be accessible and has a global object variable, **objcountry**, already declared.

Do NOT delete or change any provided code.

Follow the instructions below to code the solution.

2.2.1 Button Q2.2.1

The user will enter a country name and a leader name in the provided edit boxes named **edtcountry** and **edtleader**.

Write code to do the following:

- Instantiate the object, **objcountry**, using the country name and leader name.
- Enable the panel named **pnIQ2**.
- Load the picture named **Trees.jpg** into the image component named **imgtrees**.

(6)

2.2.2 Button Q2.2.2

Write code to receive the input from the spinedits named **sedtrees** and **sedassistants**.

Use the methods of the class to do the following:

- Increase the number of trees and the number of assistants using the values from the spinedits and calculate the amount of funds required.
- Display the country and leader, as well as the number of trees, assistants and the amount of funds required, in the label named **lbldisplay**.

Example of output:

Joseph Mcuba as leader in South Africa starting with 30 trees and 5 assistants:

Trees	30	Q2.2.2	South Africa
Assistants	5		Joseph Mcuba and 5 assistants
			30 trees
			Funds: R810,00

5 trees and 1 assistant added to the South African team with Joseph Mcuba as leader:

Trees	5	Q2.2.2	South Africa
Assistants	1		Joseph Mcuba and 6 assistants
			35 trees
			Funds: R920,00

(8)

- Enter your name and surname as a comment in the first line of the program file. (In both the class and the main program that uses the class)
- Save your programs.
- A printout of the code of both units may be required.

[40]

QUESTION 3: DATABASE PROGRAMMING

The database **WasteDisposal.mdb** contains the details of waste materials and disposal methods of a waste management company that manages the permanent removal of all waste for a city. The database contains two tables, namely **Material** and **Disposal**.

Table: Disposal

This table contains the methods of disposal of waste.

Field name	Data type	Description
Disposalcode	Number – Long Integer	A unique number assigned to each profile
DisposalMethod	Text (50)	The name of the method of waste disposal
Employees	Number – Long Integer	The number of people required to carry out the disposal method

Example of data in the **Disposal** table:

Disposal		
Disposalcode	DisposalMethod	Employees
1	Recycled	120
2	Reused	320
3	Landfill	11
4	Recoveredforenergy	24
5	Compostedonsite	4
6	Compostedoffsite	10

Table: Material

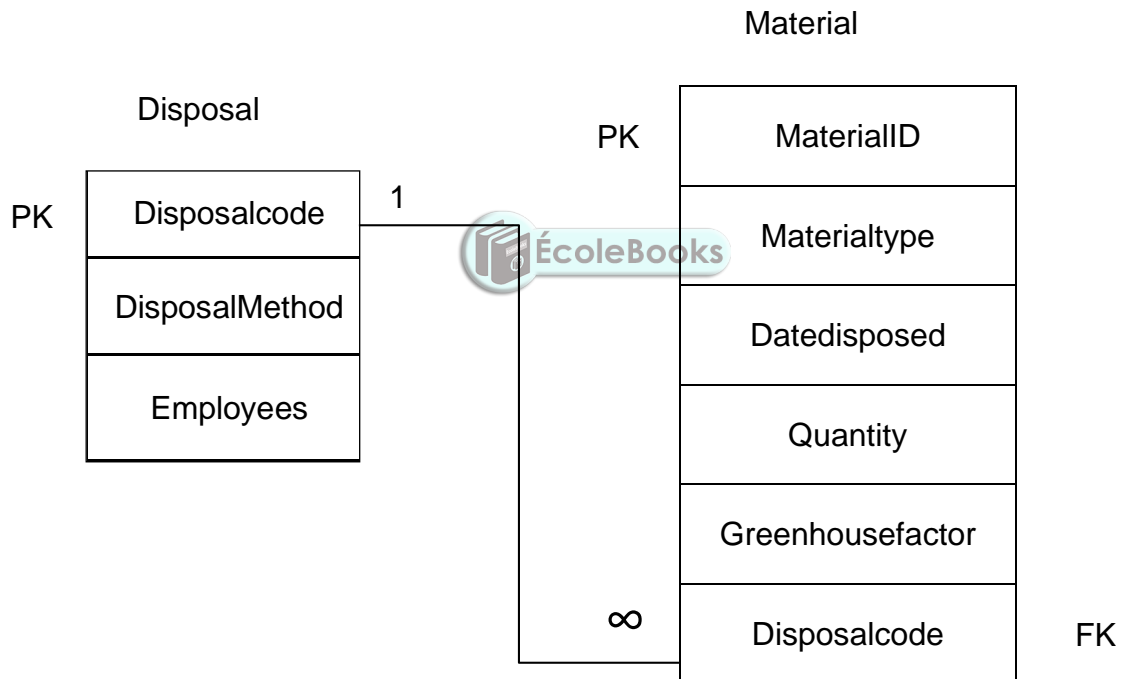
This table contains information of all the waste materials that have been disposed of by the waste management company.

Field name	Data type	Description
MaterialID	Autonumber	A unique number assigned to the material
Materialtype	Text (50)	The different types of waste materials
Datedisposed	Date/Time	Date that the last disposal took place
Quantity	Number (long integer)	Weight in Tons that was last disposed on the date described in the Datedisposed field
Greenhousefactor	Number (Double)	Carbon dioxide gas emissions factor for landfills. This number will be multiplied by the tonnage of the material, which results in the total carbon dioxide emission
Disposalcode	Number (long integer)	Foreign key to connect to the Disposal table

Example of data of the first ten records of the **Material** table.

Material					
MaterialID	Materialtype	Datedisposed	Quantity	Greenhousefactor	Disposalcode
1	Plasticcontainers	2019/10/02	85	0	1
2	Plasticbags	2019/09/28	26	0	1
3	Polystyrene	2019/07/31	45	1,8	3
4	Glassbottlesbroken	2019/09/18	74	0	1
5	Glassbottlesnotbroken	2019/07/10	25	0	2
6	Aluminiumcanswhole	2019/10/27	22	0	1
7	Aluminiumcansflattened	2019/08/13	14	0	1
8	AluminiumcansBales	2019/10/02	56	0	1
9	Steelcanswhole	2019/08/11	98	0	1
10	Steelcansflattened	2019/10/23	75	0	1

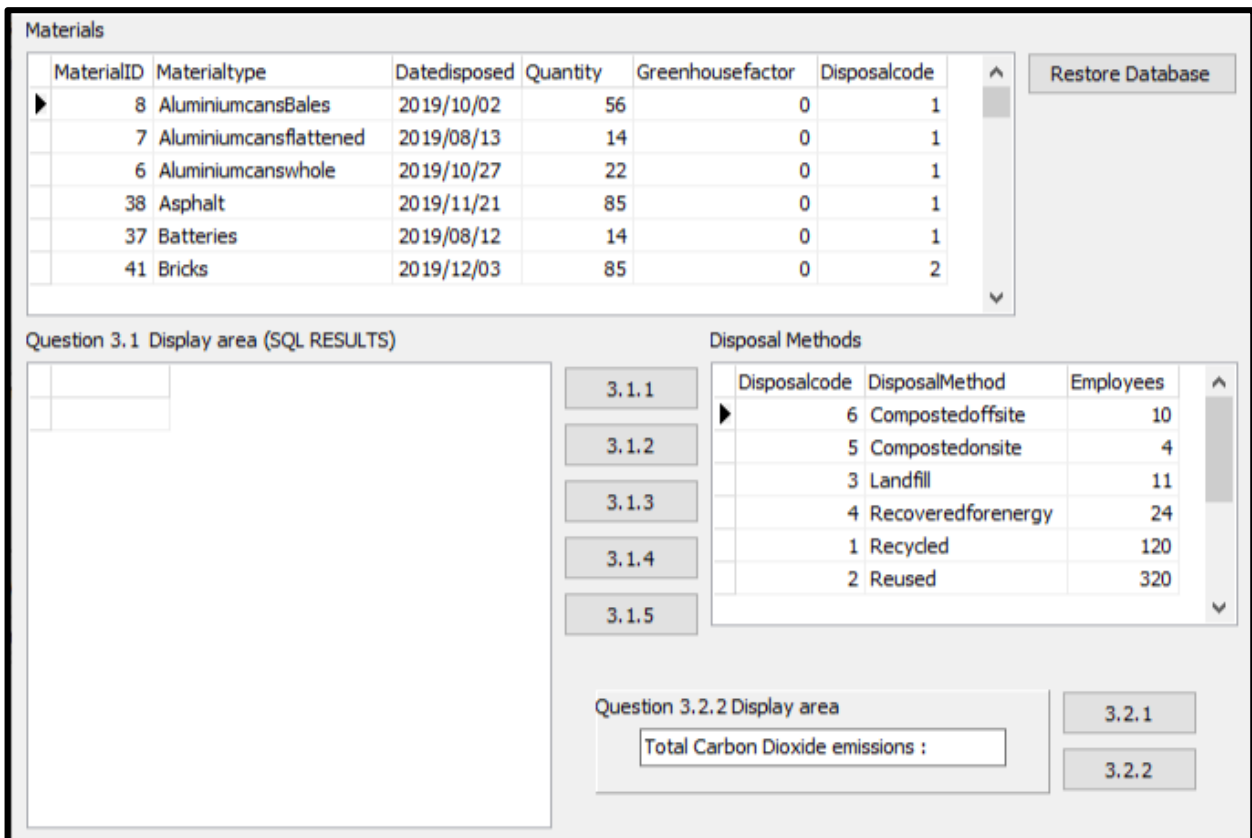
The following one-to-many relationship with referential integrity exists between the two tables in the database:



Do the following:

- Open the incomplete project file called **Question3_p.dpr** in the Question 3 folder.
- Enter your name and surname as a comment in the first line of the **Question3_u.pas** unit file.
- Compile and execute the program. The program has no functionality currently.

The user interface for all questions is displayed below:



- Follow instructions to complete the code for each question, as described in QUESTION 3.1 and QUESTION 3.2.
- **Use SQL statements to answer QUESTION 3.1 and Delphi code to answer QUESTION 3.2.**

NOTE:

- The **[Restore Database]** button is provided to restore your data contained in the database to the original content. If you need to test your code on the original data, you may click this button to restore data.
- Do NOT change any of the code provided.
- Code is provided to link the GUI components to the database.
- **TWO** variables are declared as global variables, as described in the table below.
- **Use tbldisposal and tblmaterial components in Question 3.2 only.**

Variable	Data type	Description
tbldisposal	TADOTable	Refers to the table named Disposal
tblmaterial	TADOTable	Refers to the table named Material

3.1 In this section you may ONLY use **SQL statements** to answer QUESTION 3.1.1 to QUESTION 3.1.5.

Code to execute the SQL statements and display of the results of the queries is provided. The SQL statements are incomplete.

Do the following to complete the incomplete **SQL statements** assigned to the variables sSQL1, sSQL2, sSQL3, sSQL4 and sSQL5 per question respectively.

3.1.1 Button [3.1.1]

Write SQL code to display the **MaterialType** of all materials sorted in reverse alphabetical order.

Example of output of the first 13 records:

Question 3.1 Display area (SQL RESULTS)	
MaterialType	
Woodmixed	
Tyres	
Trees	
Timber	
Tiles	
Steelcanswhole	
Steelcansflattened	
SteelcansBales	
Sawdust	
Sand	
Rubble	
Rubber	
Postswooden	

(4)

3.1.2 Button [3.1.2]

The user must enter the name of a material. Code has been provided for the material name "**wood**" to be entered in an input box and saved in a variable named **sline**.

Write SQL code to display the **Materialtype** of all materials that contain the word stored in the variable **sline**.

Example of output:

Question 3.1 Display area (SQL RESULTS)	
MaterialType	
Woodmixed	
Furniturewooden	
Fencingwooden	
Postswooden	

(5)

3.1.3 **Button [3.1.3]**

Write an SQL statement to display the **Datedisposed** and **Materialtype** of all materials disposed in October. (The year 2019 is the only year recorded in the database.)

Example of output:

Question 3.1 Display area (SQL RESULTS)	
Datedisposed	MaterialType
2019/10/02	Plasticcontainers
2019/10/27	Aluminiumcanswhole
2019/10/02	AluminiumcansBales
2019/10/23	Steelcansflattened
2019/10/14	Organicfood
2019/10/07	Grass
2019/10/06	Rubble
2019/10/16	MotorOil
2019/10/17	Concrete
2019/10/21	Cement
2019/10/23	Clinicalwaste
2019/10/08	Linoleum

(5)

3.1.4 **Button [3.1.4]**

The waste management company is expecting an increase of 10% in the number of employees next year.

Write an SQL statement to add 10% to the number of **Employees**. Rounding of numbers is not required.

(Code has been written to display the successfully updated table after editing.)

Example of output:

Question 3.1 Display area (SQL RESULTS)	
Employees	DisposalMethod
11	Compostedoffsite
4	Compostedonsite
12	Landfill
26	Recoveredforenergy
132	Recycled
352	Reused

(4)

3.1.5 **Button [3.1.5]**

Display the total of all quantities of waste materials (**Quantity**) as **Total_Quantities** for each **DisposalMethod**.

Example of output:

Total_Quantities	DisposalMethod
138	Compostedoffsite
99	Compostedonsite
813	Landfill
197	Recoveredforenergy
809	Recycled
410	Reused

(7)

3.2 In this section, only Delphi code may be used to answer QUESTION 3.2.1 and QUESTION 3.2.2.

Use the global variables, **tblmaterial** and **tbldisposal**, provided.

NO marks will be awarded for SQL statements in QUESTION 3.2.

3.2.1 **Button [3.2.1]**

Write code to set the **Greenhousefactor** to **2** for all materials that have a **Disposalcode** equal to 6.

Example of output of the last 6 records:

MaterialID	Materialtype	Datedisposed	Quantity	Greenhousefactor	Disposalcode
9	Steelcanswhole	2019/08/11	98	0	1
47	Tiles	2019/06/27	42	0	1
13	Timber	2019/06/24	33	0,6	3
27	Trees	2019/11/08	86	2	6
16	Tyres	2019/09/09	45	2,9	3
12	Woodmixed	2019/06/11	85	0,6	3

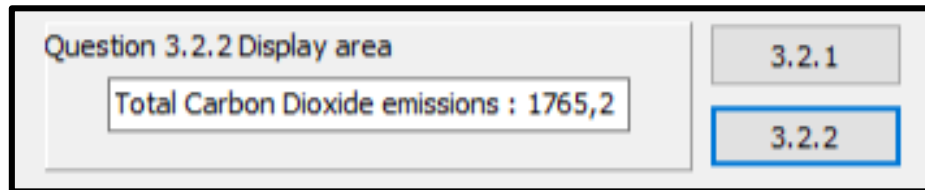
(7)

3.2.2 Button [3.2.2]

Write code to calculate the total of all carbon dioxide emissions.
CO₂ emissions are calculated by multiplying the **Greenhousefactor** by the **Quantity** (number of tons) of material waste.

Display the total by adding it to the end of the string located in the edit box named **edtdisplay**.

Example of output:



(8)

- 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

Do the following:

- Open the incomplete program in the **Question 4** folder.
- Enter your name and surname as a comment in the first line of the **Question4_u.pas** file.
- Compile and execute the program. Currently the program has no functionality.

Supplied GUI:

The supplied GUI represents the interface of the program used by a waste management company to record and report on waste disposal data.

A text file named **waste.txt** is included which contains the waste materials, method of disposal (either recycled or used as landfill) and the weights measured in kilograms of each material that was disposed of recently by the waste management company.

The format of the text file is:

<Waste material><Disposal method><weight of waste material in kg>

Example of some of the lines in the text file named **waste.txt**:

```
Paper;Recycled#2543
Cardboard;Recycled#495
Trash;Landfill#1885
Cardboard;Recycled#398
Trash;Landfill#1657
Paper;Recycled#2754
Cardboard;Recycled#769
Trash;Landfill#2025
Timber;Landfill#2125
Pallets;Recycled#398
Rubber;Recycled#4321
Tyres;Landfill#1258
Metal;Recycled#541
Food;Landfill#774
Grass;Landfill#1142
Trees;Landfill#685
Soil;Landfill#5547
Rubble;Recycled#3984
Clay;Recycled#547
```



A constant array named **arrtypes** has been declared and filled with unique waste material types.

```
arrtypes: array[1..15] of string =
('Paper','Cardboard','Trash','Timber','Pallets','Rubber','Tyres','Metal','Food','Grass','Trees',
'Soil','Rubble','Clay','Computers');
```

Code has been written to read the items from each line of the text file into 2 global, parallel arrays, named **arrdata** and **arrweights**.

arrdata : array[1..100] of string;
arrweights: array[1..100] of integer;
icount: integer;

Arrdata contains the name of each waste material joined to either the word *Recycled* or *Landfill* with a semicolon separating the two words.

Arrweights contains integers representing the weight of the corresponding waste material in kilograms.

The variable **icount** holds the total number of items for the two parallel arrays.

Code has been written in the **OnActivate** event of the form to read data from the text file into the two arrays and then display the array contents into the rich edit.

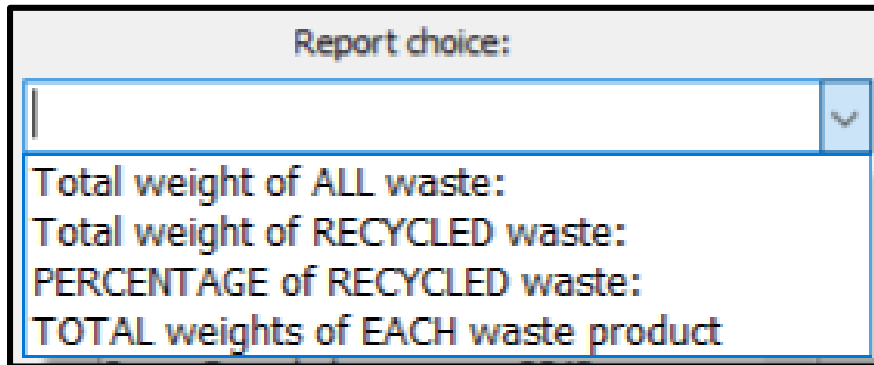
Weights of waste products in Kgs	
Paper;Recycled	2543
Cardboard;Recycled	495
Trash;Landfill	1885
Cardboard;Recycled	398
Trash;Landfill	1657
Paper;Recycled	2754
Cardboard;Recycled	769
Trash;Landfill	2025
Timber;Landfill	2125
Pallets;Recycled	398
Rubber;Recycled	4321
Tyres;Landfill	1258
Metal;Recycled	541
Food;Landfill	774
Grass;Landfill	1142
Trees;Landfill	685
Soil;Landfill	5547
Rubble;Recycled	3984
Clay;Recycled	547
Computers;Recycled	6541
Paper;Recycled	998
Cardboard;Recycled	589
Trash;Landfill	6854
Cardboard;Recycled	1542
Trash;Landfill	1441
Paper;Recycled	1698
Cardboard;Recycled	885

Complete the code for each question, QUESTION 4.1 and QUESTION 4.2.

- NOTE:**
- Good programming techniques and modular design must be applied in the design and coding of your solution.
 - You may NOT change the code provided.

You must write code in the **onchange** event of the combobox named **cmbreport**.

Below is a diagram of the 4 options of the combobox.

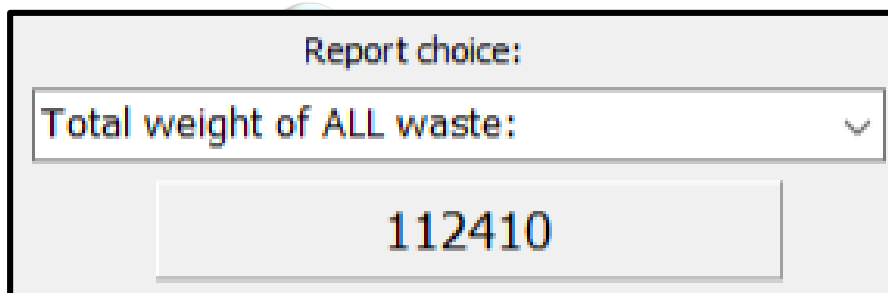


For QUESTIONS 4.1 to 4.4 you must write code for each of the combobox options.

4.1 **Combobox option [Total weight of ALL waste]**

Calculate and display the total weight of all waste materials collected. The total must be displayed on the panel named **pnloutput**.

Example of output:



(4)

4.2 **Combobox option [Total weight of RECYCLED waste]**

Calculate and display the total weight of all recycled waste material. The total must be displayed on the panel named **pnloutput**.

Example of output:



(5)

4.3 Combobox option [Percentage of RECYCLED waste]

The recycled weight must be divided by the total weight and then multiplied by 100 and rounded to a whole number, in order to calculate the percentage of waste that was recycled.

Display the result on the panel named **pnloutput**.

Example of output:

The screenshot shows a window titled 'Report choice:'. Inside, there is a combobox menu with the text 'PERCENTAGE of RECYCLED waste:' and a dropdown arrow. Below the combobox is a text box containing the number '46'.

(3)

4.4 Combobox option [TOTAL weight of EACH waste product]

The array named **arrdata** has many waste types that are duplicated, eg 'Cardboard;Recycled' occurs many times in the array.

The global constant array named **arrtypes** contains all of the unique types of waste that have been disposed of by the waste management company.

Use this information to calculate the total weights collected for each unique waste type and write the results to a text file named **report.txt**.

(Note: The text file named **report.txt** must be created using Delphi code.)

Example of the output to the text file, **report.txt**:

```
Paper = 13332
Cardboard = 6669
Trash = 22606
Timber = 5488
Pallets = 1194
Rubber = 5843
Tyres = 4643
Metal = 1624
Food = 4052
Grass = 5650
Trees = 11801
Soil = 7010
Rubble = 4872
Clay = 4986
Computers = 12640
```

(18)

- 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