higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATE (VOCATIONAL)

PRINCIPLES OF COMPUTER PROGRAMMING

(Second paper)

NQF LEVEL 3

(10041023)

11 November 2023 (X-paper)
09:00–13:00

Candidates are allowed 15 minutes to read the instructions before the start of the examination.

This question paper consists of 19 pages.

503Q2N2217

Copyright reserved

TIME: 4 HOURS
MARKS: 100

INSTRUCTIONS AND INFORMATION

1. Candidates must be seated in the computer laboratory 45 minutes before the start of the examination to ensure that the allocated computer has the appropriate software and can make PRINTOUTS.

2. Answer all the questions.

3. Read all the questions carefully.

4. Number the answers according to the numbering system used in this question paper.

5. This question paper is set with programming terms that are specific to the BASIC programming language and must be completed using Visual Studio.NET or VB.NET Express together with MS Office or applicable word processing software.

6. Answer the questions according to the instructions and specifications given in each question. Marks will be awarded according to the set requirements.

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

8. Routines such as *search*, *sort*, *selection* and *file handling* must be hard coded by you. You may not use the built-in features of the programming language.

9. All data structures must be defined by you, unless the question contains partially completed source code.

10. If a question contains partially completed source code, do not delete the source code.

11. Candidates are allowed to use the help function of the integrated development environment.

12. Work must be saved regularly on the hard disk/flash disk/network drive that you have been given, to prevent loss of work. No additional time may be allowed to redo lost work that has not been saved.

Copyright reserved

13. In the event of problems such as a power failure, computer or printer breakdown, the invigilator will make the necessary arrangements to continue with the examination session. Candidates will not be penalised with time lost under such circumstances.

14. No candidate may print their work or make a flash drive available for another candidate or access another candidate's work via the network. Any attempt to access any information from or transfer information to another candidate in whatever form is a contravention of the examination rules and regulations and will be regarded and managed as such.

15. Candidates must ensure that they print their work immediately after they have completed a question. The candidate's EXAMINATION NUMBER must appear on all PRINTOUTS. No questions without a printed EXAMINATION NUMBER will be marked. Source code PRINTOUTS and screenshot PRINTOUTS with handwritten examination numbers will not be accepted or marked.

16. The source code must be printed directly from the integrated development environment (IDE). Candidates are not allowed to copy the source code from the IDE and paste it in MS Word for printing purposes.

17. All PRINTOUTS of the source code and screenshots printed during the examination session must be handed in to the invigilator.

18. Steps to print a user interface or any other screen shot:

    18.1 Run/Debug the solution (for Visual Studio.NET ONLY).

    18.2 Hold down the ALT key and press the print screen (Prnt Scrn) key.

    18.3 Open a blank MS Word document and paste the screen shot.

    18.4 Insert a blank THREE-column header.

        (a) Add your EXAMINATION NUMBER against the left margin.

        (b) Add the QUESTION NUMBER against the right margin.

    18.5 Make a PRINTOUT of the page.

19. The following VB.NET solution files have been saved on the computer in the PCP L3 P2 *Examination* folder:

    19.1 QUESTION 1 → …*PCP L3 P2*\ *Question_1**Question_1.sln*

    19.2 QUESTION 2 → …*PCP L3 P2*\ *Question_2**Question_2.sln*

    19.3 QUESTION 3 → …*PCP L3 P2*\ *Question_3**Question_3.sln*

    19.4 QUESTION 4 … No files

Copyright reserved

QUESTION 1: GENERAL PROGRAMMING SKILLS

The following questions refer to the VB.NET solution called *Question_1* that has been saved on the workstation.

Mzansi Superstore has recently launched their loyalty rewards programme for customers. Customers who sign up and purchase at the store will receive a certain number of loyalty points. Mzansi Superstore requires assistance in writing a VB.Net application that will assist with the loyalty programme.

FIGURE 1 below is a screen print of what the graphical user interface looks like. The completed interface has been saved on the workstation.

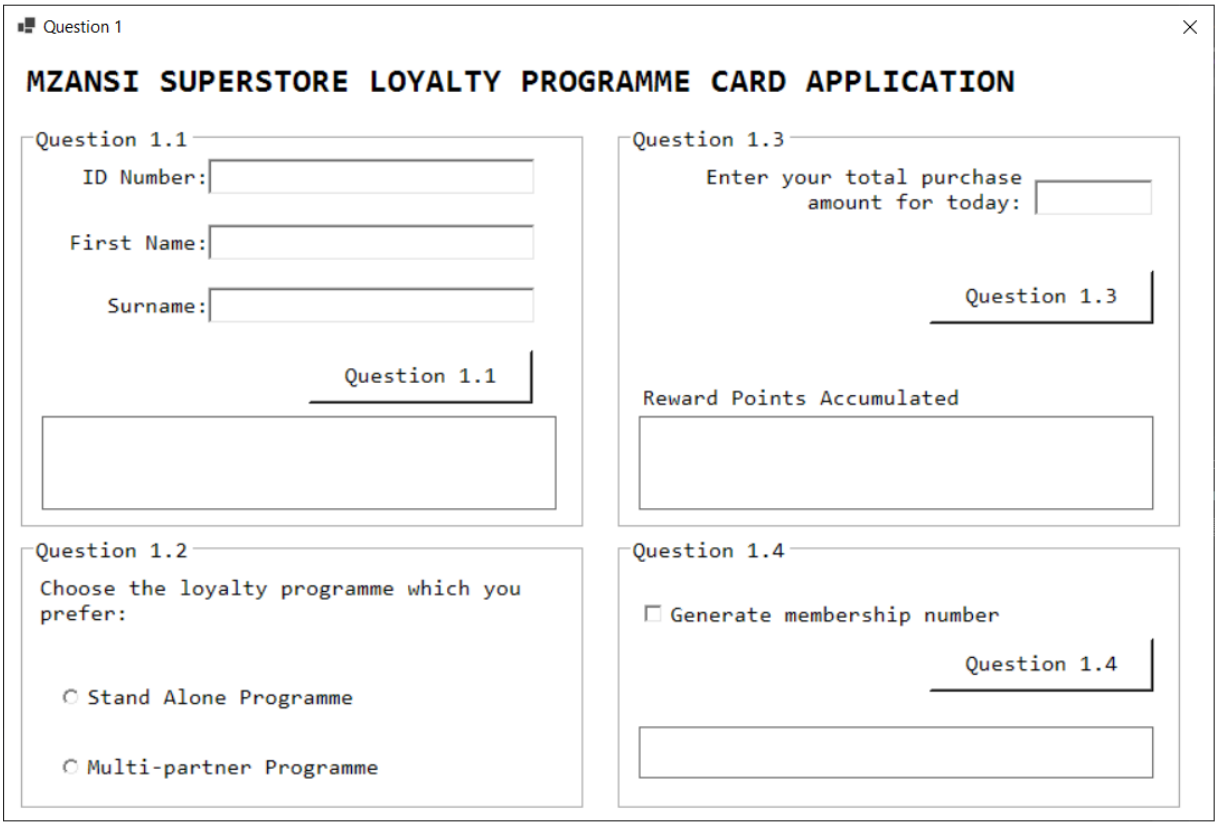


FIGURE 1: SCREENSHOT OF INTERFACE

The user interface displays FOUR (4) different sections named Question 1.1 to Question 1.4. The completed graphical user interface (GUI), with partially completed source code, has been saved on the computer.

Open the VB.NET solution *Question_1.sln* and follow the instructions below to complete the application.

Copyright reserved

1.1 When the **QUESTION 1.1** button is clicked:

1.1.1 Write code to validate the identity number entered by the user as follows:

- The identity number cannot be blank/empty.

If the input is not valid, the application must draw the user's attention as follows:

- Change the colour of the identity number text box to red.
- Move the cursor to the identity number text box.
- Terminate the current procedure.

Example of output if validation is not successful:



(4)

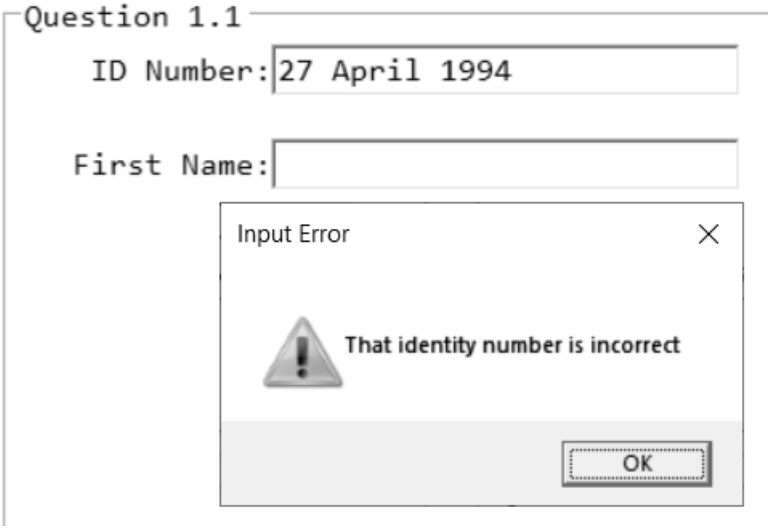
1.1.2 Write code to validate that a correctly formatted identity number has been entered by the user as follows:

- The identity number should consist of numbers only.
- The identity number cannot be shorter or longer than thirteen numbers.

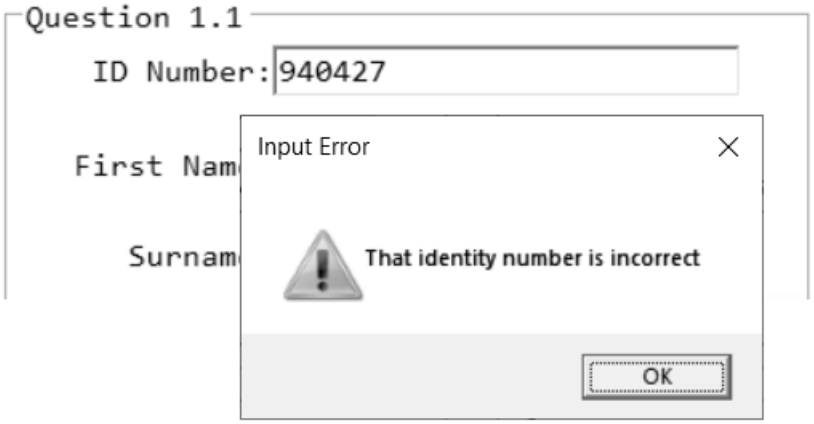
If the input is valid, assign the identity number to the provided global variable.

If the input is not valid, the application must draw the user's attention by displaying a pop-up dialog box, moving the cursor to the identity number textbox and terminating the procedure. (4)

Example of output if the identity number does not contain numbers:



Example of output if the identity number is shorter/longer than thirteen numbers:

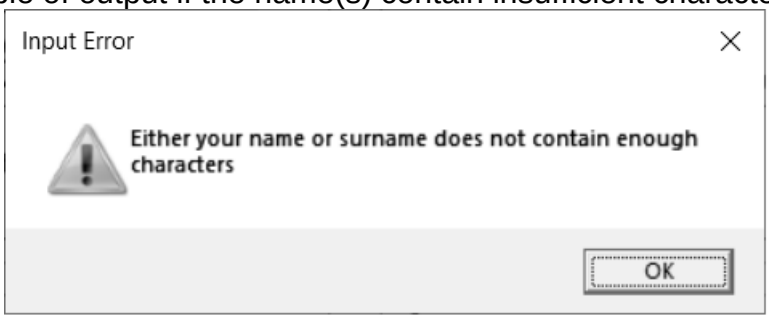


1.1.3 Write code to validate an appropriate length for a valid name and surname as follows:

- The first name and surname must consist of two or more characters.

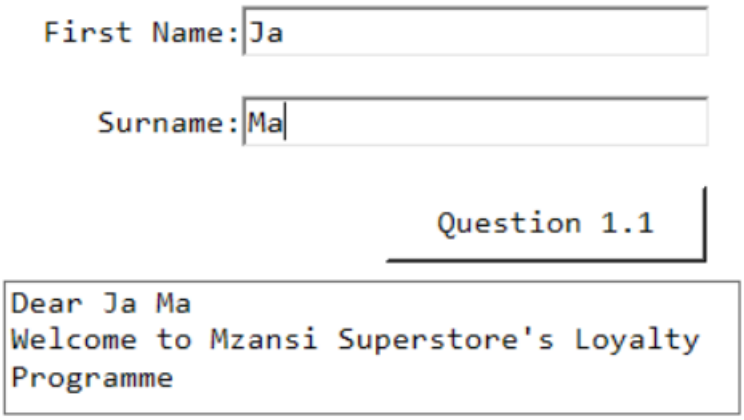
If the input is not valid, the application must draw the user's attention by displaying a pop-up dialog box to indicate the specific error. (3)

Example of output if the name(s) contain insufficient characters:



1.1.4 Write code to concatenate the name and surname of the customer and display it in the label. (2)

Example of welcome message:



1.2 The customers of Mzansi Superstore have the choice of joining two loyalty programmes, as indicated by the radio buttons on the interface.

- A stand-alone programme.
- A multi-partner programme.

Write code so that when the user selects the stand-alone programme, they are credited with 1 000 points, and 500 points for the multi-partner programme. The points must be assigned to the global variable, *intPoints*. (2)

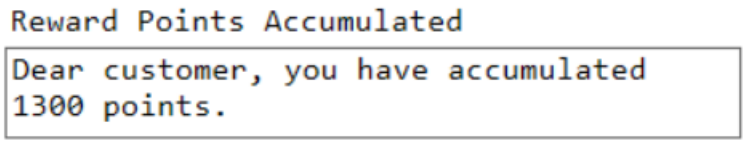
1.3 When the **QUESTION 1.3** button is clicked:

The total purchase amount must be entered into the text box.

Write code so that the application will calculate how many loyalty points will be allocated to the customer, in addition to the loyalty points calculated in QUESTION 1.2. The application must calculate and display a message indicating the total loyalty points the customer has accumulated according to the following criteria.

- A customer will earn ONE loyalty point for every FIVE rand spent shopping at the store. (4)

Example of output if the customer is part of the stand-alone loyalty programme and has purchased for an amount of R1 500:



1.4 When the **QUESTION 1.4** button is clicked:

If the *Generate Membership Number* check box is selected:

Write code so that the application will generate a membership number based on the following information:

The customer's ID number entered in QUESTION 1.1 and the type of loyalty programme selected in QUESTION 1.2 are required to compute the customer's membership number. The membership number is made up from two parts.

The first part of the membership number is based on the selection made in QUESTION 1.2 and can either be the character, S, if the stand-alone programme was selected or the character, M, if the multi-partner programme was selected, followed by the first six digits of the customers ID Number.

The second part of the membership number is determined by the current date in the format DDMMYYYY, which is two-digit day, two-digit month, four-digit year (for example: 27 April 2022 → 27/04/2022).

(6)

Example of output based on the following customer:

Name: John, Surname: Dube, ID Number: 9404275050081
Type of loyalty programme: Multi-partner.

```
M94042707/10/2021
```

1.5
- Enter your EXAMINATION NUMBER, COMPUTER WORKSTATION NUMBER and DATE as comments in the first three lines of the source code.
- Save the source code.
- Make a PRINTOUT of the source code.

[25]

QUESTION 2: MODULAR PROGRAMMING (ARRAYS AND PROCEDURES)

The following questions refer to the VB.NET solution called *Question_2* that has been saved on the workstation.

Mzansi Fuels has decided to create an application to help their customers track their fuel consumption on a weekly (seven days) basis. Fuel consumption is the measure of the amount of fuel it takes to travel a certain distance and is measured in litres per 100 kilometres (ℓ/100). Mzansi Fuels require assistance in completing the VB.Net application.

FIGURE 2 and FIGURE 3 below is a screen print of what the graphical user interface looks like. The completed interface has been saved on the workstation.

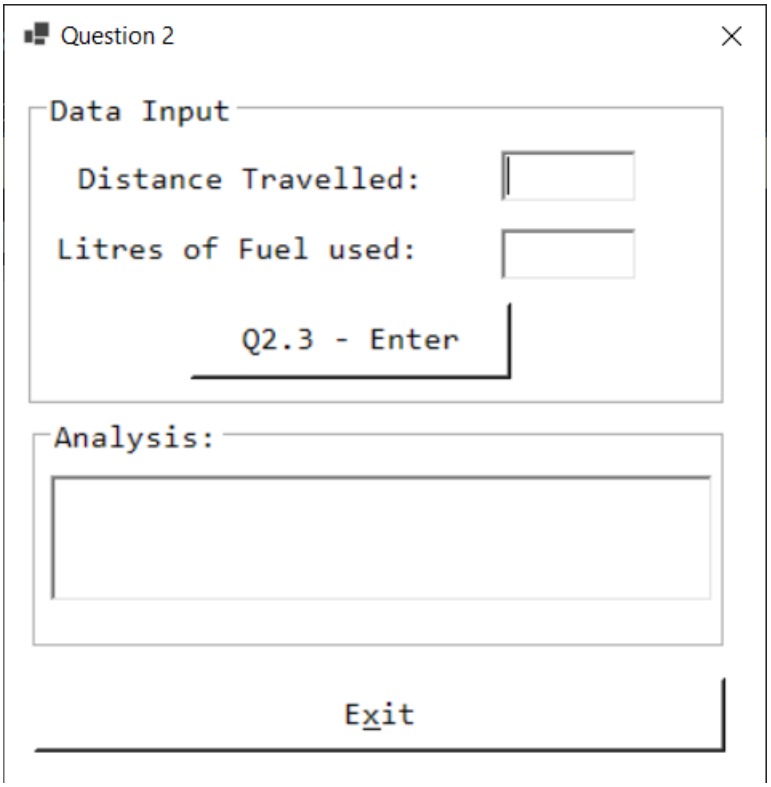


FIGURE 2: SCREEN SHOT OF INTERFACE

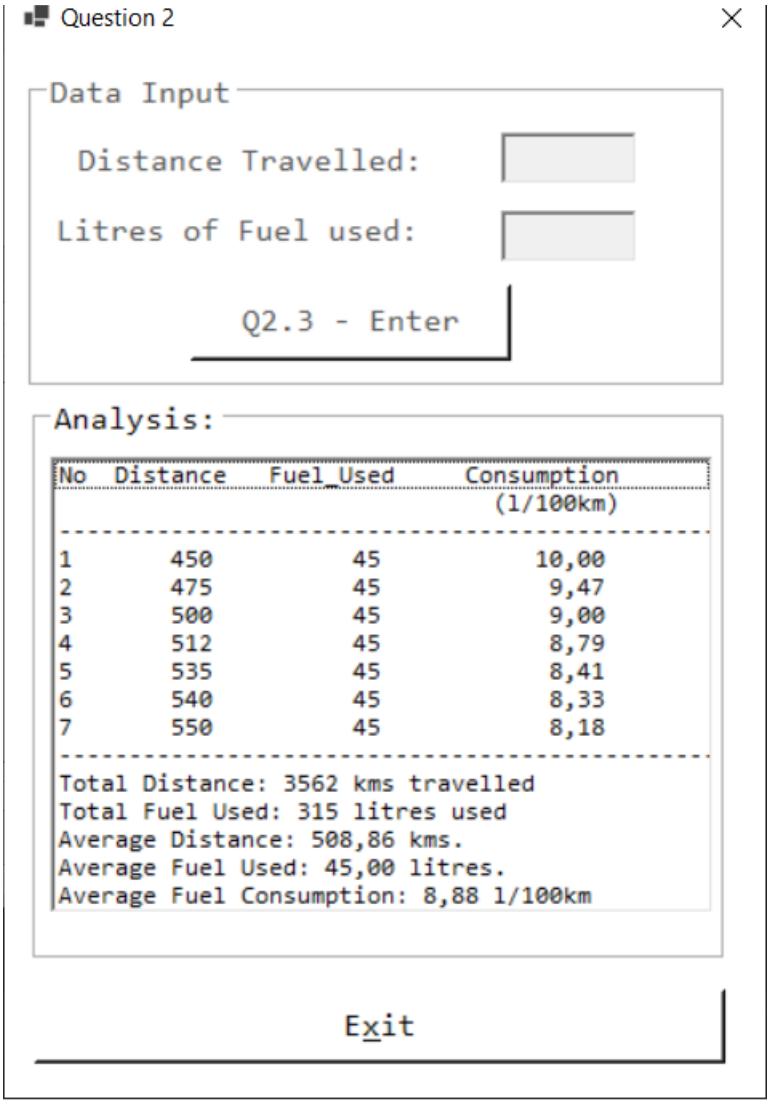


FIGURE 3: SCREEN SHOT OF INTERFACE

The user interface displays TWO sections, one for the data input and the other for the analysis of the fuel consumption. The VB.Net application has been partially coded.

Open the VB.NET solution, *Question_2.sln*, and follow the instructions below to complete the application.

2.1 Write code to create a loop counter variable, and THREE parallel, single dimension arrays to store SEVEN items of data for the following:

- Distance Travelled.
- Litres of Fuel used.
- Fuel consumption.

(4)

HINT: See FIGURE 3.

2.2 Write code to create a user-defined function procedure called *CalcConsumption*, which will receive copies of the distance travelled and the fuel used as parameters. The procedure must calculate and return the fuel consumption based on the following formula.

*Fuel consumption = Fuel used / Distance travelled * 100* (7)

NOTE: Apply BODMAS rule.

2.3 When the **QUESTION 2.3** button is clicked:

2.3.1 Write code to assign the contents of both text boxes into the relevant parallel array. (3)

2.3.2 Write code to increment the loop counter variable. (1)

2.3.3 Write code to determine if the maximum limit of the array has been exceeded.

(a) If TRUE, terminate the procedure. (2)

(b) If FALSE:

- Prevent the user from entering any more data.
- Clear the list box.
- Create headings in the list box as shown in the output below.

```
No  Distance    Fuel_Used      Consumption
                                (1/100km)
-------------------------------------------
```

HINT: Use an appropriate method to align the characters to the left and right sides of the list box.

- Iterate through the distance travelled array to calculate the fuel consumption for each of the seven days, and display the result in the list box as shown in the output below.

```
Analysis:
No  Distance    Fuel_Used      Consumption
                                (1/100km)
-------------------------------------------
1       489          40            8,18
2       500          40            8,00
3       425          40            9,41
4       475          40            8,42
5       477          40            8,39
6       423          40            9,46
7       452          40            8,85
-------------------------------------------
```

- Declare THREE (3) variables for the total distance travelled, the total fuel used and the total fuel consumption.
- Iterate through the distance travelled array to calculate values for the total distance travelled, the total fuel used and the total fuel consumption, and assign to the variables created in the previous step.
- Display the total values for distance travelled and fuel used in the list box, and calculate and display the average distance travelled, the average fuel used and the average fuel consumption as shown in the output below.

```
Analysis:
No  Distance    Fuel_Used      Consumption
                                (1/100km)
-------------------------------------------
1       489          40            8,18
2       500          40            8,00
3       425          40            9,41
4       475          40            8,42
5       477          40            8,39
6       423          40            9,46
7       452          40            8,85
-------------------------------------------
Total Distance: 3241 kms travelled
Total Fuel Used: 280 litres used
Average Distance: 463,00 kms.
Average Fuel Used: 40,00 litres.
Average Fuel Consumption: 8,67 l/100km
```

(18)

Copyright reserved

2.4

- Enter your EXAMINATION NUMBER, COMPUTER WORKSTATION NUMBER and DATE as comments in the first three lines of the source code.
- Save the source code.
- Make a PRINTOUT of the source code.

[35]

QUESTION 3: FILE AND ERROR HANDLING

The following questions refer to the VB.NET solution called *Question_3* that has been saved on the workstation.

The English (FAL) lecturer at your college has requested you to assist in completing an application for the NCV level 2 students. The application loads a text file that contains words. The student is then able to enter letters and search for a word.

FIGURE 4 below is a screen print of what the graphical user interface looks like. The completed interface has been saved on the workstation.



FIGURE 4: SCREENSHOT OF INTERFACE

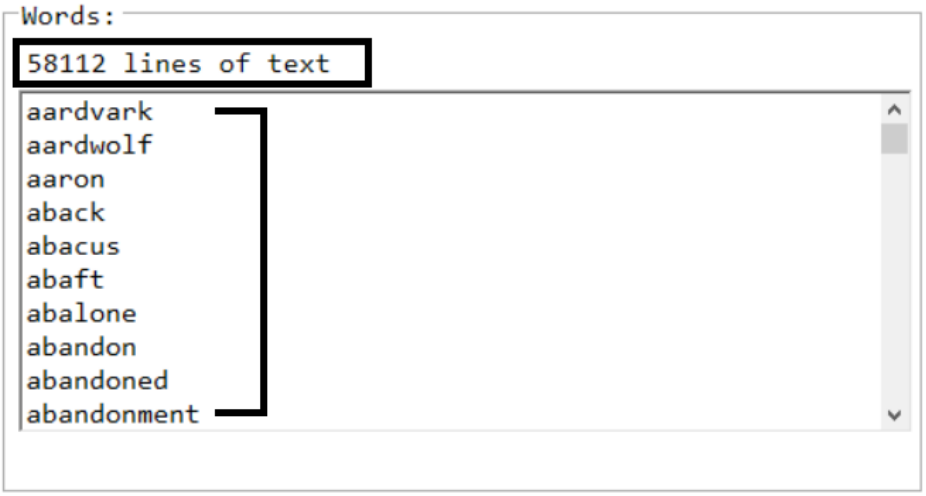
The user interface displays TWO sections, one for the data input of letters and the other for the displaying the word or words that are found. The VB.Net application has been partially coded.

Open the VB.NET solution, *Question_3.sln*, and follow the instructions below to complete the application.

3.1 When the application loads:

3.1.1 Write exception handling code to populate the list box with the contents of the text file *dictionary.txt*. Handle any exceptions by displaying a pop-up dialog box with a relevant message. The application must also determine the number of lines in the text file. (11)

Example of output:



3.2 When the **QUESTION 3.2** button is clicked:

3.2.1 Write code that will use an appropriate method to find a word from the list of words in the list box based on the input provided in the text box, and assign the result to a new integer variable. (2)

3.2.2 Write code to highlight the corresponding word in the list box based on the integer variable created in QUESTION 3.2.1. (2)

Example of output:



3.3
- Enter your EXAMINATION NUMBER, COMPUTER WORKSTATION NUMBER and DATE as comments in the first three lines of the source code.
- Save the source code.
- Make a PRINTOUT of the source code.

[15]

QUESTION 4: DATABASE

The supply chain manager at your college has approached you for assistance in creating a prototype database that the college can use to keep track of their suppliers and the products they offer.

The database needs to store information in two tables (SUPPLIER and PRODUCT).

Below is the data specification of the two tables:

FIELD NAME	DATA TYPE	FIELD SIZE	INPUT MASK	DESCRIPTION
SupplierID	Number	Long Integer	N/A	6-digit unique supplier identification code Primary Key (PK)
SupplierName	Short Text	22	N/A	Supplier's name
SupplierContact	Short Text	12	Telephone e.g.: (011)854-1246	Suppliers telephone number

TABLE 1: SUPPLIER

FIELD NAME	DATA TYPE	FIELD SIZE	FORMAT	LOOKUP VALIDATION	DESCRIPTION
ProdID	Short Text	8	N/A	N/A	8-character alphanumeric unique code (PK)
ProdDesc	Short Text	40	N/A	N/A	Description of the product
UnitPrice	Currency	N/A	N/A	N/A	Price per unit
Discount	Single	N/A	%	Maximum of 25%	Discount percentage
SupplierID	Number	Long Integer	N/A	SupplierID	Foreign key

TABLE 2: PRODUCT

The database application needs to adhere to the following business rule:

- A product can be supplied by one or more suppliers.

Copyright reserved

4.1 Use the information above to set up the database in MS Access. Open a new blank database and save it as Supplier together with your EXAMINATION NUMBER. For example: Supplier – 940427XXXX081.

4.1.1 Create the Supplier table (refer to TABLE 1) and save it as Supplier.

4.1.2 Create a Product table (refer to TABLE 2) and save it as Product.

4.1.3 Create an appropriate relationship between the SUPPLIER and PRODUCT tables. Enforce referential integrity.

4.1.4 Open the supplier table in design view. Select the supplier contact field.

Make a screen shot of the table design and print it. (2)

4.1.5 Open the product table in design view. Select the discount field.

Make a screen shot of the table design and print it. (2)

4.1.6 Use the database documenter to set the print table definition options as shown in FIGURE 5 below:

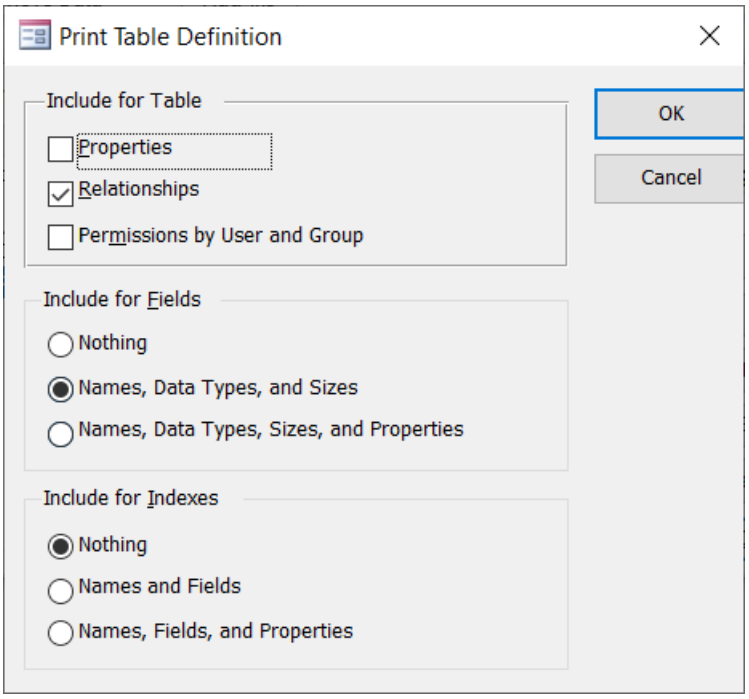


FIGURE 5: TABLE STRUCTURE SETTINGS

Print the database table structures for both tables. (8)

4.2 Open the Supplier table. Add the following information to the table:

ID	NAME	CONTACT NUMBER
210214	AKIHB Investments	011 567 8000
202154	ButaMax Fuels	015 645 6000
195456	Mbuso's House of Wood	013 456 3434

Save the records and close the table.

Add the following products to the different suppliers.

ID	DESCRIPTION	PRICE	DISCOUNT	SUPPLIER
PT001	Fire suppressant	R200,00	10%	AKIHB Investments
PT002	9 kg LPG Cylinder	R280,00	15%	ButaMax Fuels
PT003	Vintage School Desk	R785,99	25%	Mbuso's House of Wood
PT004	Laminated glass	R214,00	14%	AKIHB Investments
PT005	13 kg LPG Cylinder	R985,00	22%	ButaMax Fuels
PT006	FDB Lab Stool	R349,99	5%	Mbuso's House of Wood

Save the records and close the table.

4.3 Create a database query using both tables.

The query should produce the following output:

SupplierName	ProdDesc	Price
ButaMax Fuels	9Kg LPG Cylinder	280
Mbuso's House of Wood	Vintage School Desk	786
ButaMax Fuels	13Kg LPG Cylinder	985
Mbuso's House of Wood	FDB Lab Stool	350

The query must display the prices of records where the unit price exceeds R250 as whole numbers.

Save the query as Price Query with your EXAMINATION NUMBER.

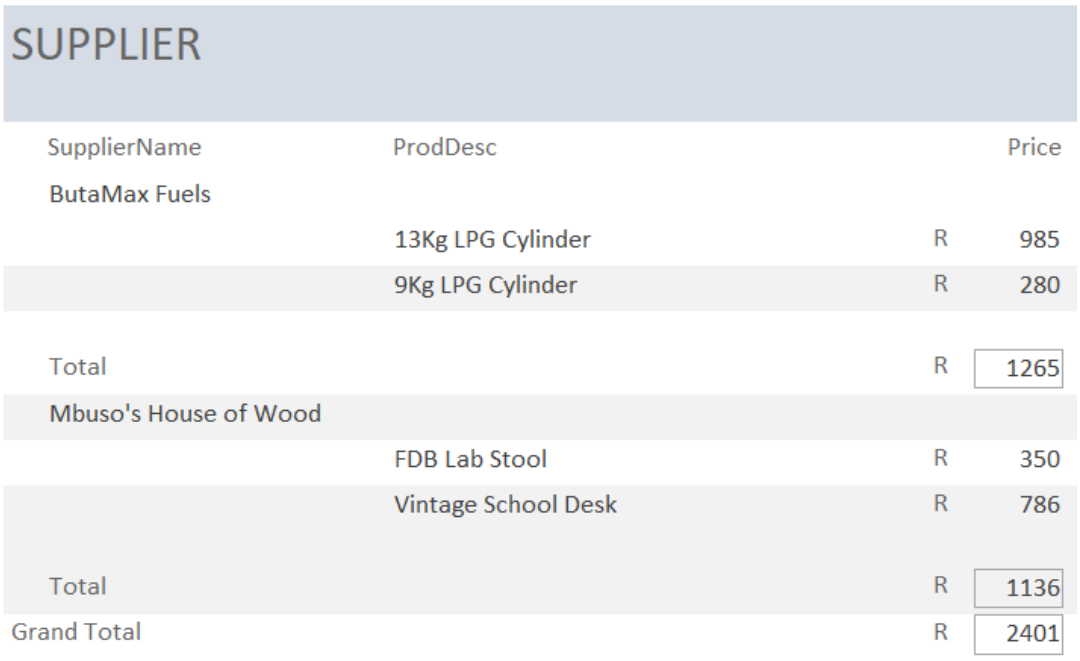
4.3.1 Open the query in design view, ensure that all columns are clearly visible and make a screen print of the query form in design view. (5)

4.3.2 Run the query and print the resulting records. (2)

Copyright reserved

4.4 Create the following database report using the query created in QUESTION 4.3.

The report should produce the following output:



SUPPLIER

SupplierName	ProdDesc		Price
ButaMax Fuels			
	13Kg LPG Cylinder	R	985
	9Kg LPG Cylinder	R	280
Total		R	1265
Mbuso's House of Wood			
	FDB Lab Stool	R	350
	Vintage School Desk	R	786
Total		R	1136
Grand Total		R	2401

Group the report detail on the different categories.

The report should also calculate and display the total price for all the listed products.

Save the report as Price Report with your examination number. Ensure that your examination number appears in the report header.

Print the report. (6)

[25]

TOTAL: 100

Copyright reserved