



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

INFORMATION TECHNOLOGY P1

FEBRUARY/MARCH 2013

MARKS: 120

TIME: 3 hours

This question paper consists of 16 pages and 5 annexures.

INSTRUCTIONS AND INFORMATION

1. 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.
2. With regard to the formulation of the questions in terms of programming, no distinction has been made between the two programming languages in this question paper. Where required, specific instructions have been provided for Delphi and Java candidates respectively.
3. You require the files listed below in order to answer the questions. They are EITHER on a stiffer disk OR CD issued to you OR the invigilator/teacher will tell you where to find them on the hard drive of the workstation you are using OR in a network folder.

Question1_Delphi:

Question1P.dpr
Question1P.res
Question1U.dfm
Question1U.pas
tblTourists.txt
tblTours.txt
TourismDB.mdb

Question1_Java:

tblTourists.txt
tblTours.txt
TestQuestion1.java
Tourism.java
TourismDB.mdb

Question2_Delphi:

DataQ2.txt
Question2P.dpr
Question2P.res
Question2U.dfm
Question2U.pas
uQuest2.pas

Question2_Java:

DataQ2.txt
Quest2.java
TestQuestion2.java

Question3_Delphi:

DataQ3_Delphi.txt

Question3_Java:

DataQ3_Java.txt

If you received the files above on a disk (CD or stiffer), write your examination number on the label.

4. Type your examination number as a comment in the first line of each program file that contains your programming code.
5. Your program should always be coded to answer the question in the way it has been formulated. You are not allowed to only copy the given output supplied in the question paper.
6. Read ALL the questions carefully. Do not do more than the questions require.
7. To help you to understand each question better, you have to read the entire question before answering any subquestions.

8. Save your work at regular intervals as a precaution against power failures.
9. There might be a technical interruption that prevents you from writing the examination, such as a power failure. When you resume writing the examination, you will be given the time remaining when the interruption began, and an additional 10 minutes.
10. During the examination, you may use the manuals originally supplied with the hardware and software. You may also use the HELP functions of the software. Candidates using Java may use the Java API files. You may NOT use any other resource material.
11. At the end of this examination session, you must hand in the disk or CD with all your work saved on it OR you must make sure that all your work has been saved on the hard drive/network as explained to you by the invigilator/teacher. Ensure that all files can be read.
12. Make printouts of the programming codes of all the programming questions you have done.
13. All printing of programming questions that you have done will take place within an hour of the completion of this examination.

SCENARIO

Blue Crane Tours is a company offering sightseeing tours of South Africa to overseas visitors. The company wants to improve their business; therefore they require new software to assist them with their administration. You are required to complete the following **THREE** questions using the programming language you have studied.

QUESTION 1: PROGRAMMING AND DATABASE

A Microsoft Office Access database named **TourismDB.mdb**, two text files (**tblTourists.txt** and **tblTours.txt**) and an incomplete program are given in the folder named **Question1_XXXX** where **XXXX** refers to the programming language you have studied.

The design of the tables in the **TourismDB** database and sample data from each table are given in **ANNEXURE A**.

Do the following:

- Make a backup copy of the **TourismDB** database **BEFORE** you start answering the questions. You will need a copy of the original database to be able to test your program thoroughly.
- Rename the folder for QUESTION 1 by replacing the name of the programming language you have studied with your examination number.
- Open the incomplete program for QUESTION 1.
- Enter your examination number as a comment in the first line of the program file.
- Compile and execute the program. The interface displays eight menu options as indicated in the section labelled **QUESTION 1** in **ANNEXURE B (Delphi)/ANNEXURE C (Java)**.

NOTE:

- An error message will be displayed if any of the options A–G are selected because of the incomplete SQL statements.
- If you experience any problems using the database or connecting to the database, refer to **ANNEXURE D (Delphi)/ANNEXURE E (Java)** for troubleshooting hints.
- If you still experience database problems, you must nevertheless do the SQL code and submit it for marking. **Marks will only be awarded for the programming code that contains the SQL statements.**

- Complete the code for each menu option by formulating an appropriate SQL statement to display the respective query results as described in QUESTIONS 1.1 to 1.7 below.

NOTE: The code to some input statements and the code to execute the SQL statements and display the results of the queries have already been written as part of the given code.

1.1 Menu Option A

Display all the information of the tours in the **tblTours** table. Sort the data by the destination in alphabetical order and then by the starting date of the tour in descending order.

Example of the output of the first four records:

TourID	Surname	Initial	Destination	StartDate	EndDate	Seats	Price
23	Smith	P	Augrabies National Park	2012/03/17	2011/03/29	15	R 2 903.00
1	Khumalo	J	Augrabies National Park	2011/03/19	2011/03/24	19	R 2 903.00
25	Khumalo	J	Blyde River Canyon	2012/05/11	2012/05/18	10	R 4 926.00
19	Nel	M	Blyde River Canyon	2012/02/22	2012/03/07	25	R 4 926.00

:

NOTE: The format of the dates in the **StartDate** and **EndDate** fields and the format of the **Price** field may differ from the example above. (4)

1.2 Menu Option B

A lost suitcase that probably belongs to a female tourist was found. The suitcase has the letters CC engraved on it which may refer to the initials of the tourist's first name and surname. Display the **TourID**, the **FirstName** and **Surname** of all the female tourists, whose first names and surnames start with a C.

Example of the output:

TourID	FirstName	Surname
14	Caroline	Carrier
17	Candelora	Columbo

(4)

1.3 Menu Option C

Allow the user to enter the name of a country, for example Spain. Display the **TourID** and the **Surname** of all the tourists from that country who have paid their deposit.

Example of the output of the first four tourists from Spain (on the next page):

TourID	Surname
19	Spina
4	Di
34	Mancuso
43	Ghislieri

:

(4)

1.4 Menu Option D

Display a list of all the tour guides who guided tours that exceeded a period of 5 days during the period 12 June 2012 to 31 October 2012. Display the surname of the tour guide, the starting date, the end date and the duration of the tour (number of days). The number of days has to be calculated. Use **NumberOfDays** as the heading of the calculated field.

Example of the output of the first four records:

Surname	StartDate	EndDate	NumberOfDays
De Wet	2012/06/12	2012/06/17	6
Lemmer	2012/06/20	2012/07/02	13
Nel	2012/07/13	2012/07/21	9
Ramphopolos	2012/07/15	2012/07/20	6

:

NOTE: The format of the dates in the **StartDate** and **EndDate** fields and the format of the **NumberOfDays** field may differ from the example above.

(6)

1.5 Menu Option E

All the records of tours that began and ended during the year 2011 need to be removed from the database.

Code an SQL statement to implement the removal of the appropriate records.

Once the records have been removed successfully, a suitable message will be displayed.

(4)

1.6 Menu Option F

Calculate and display the total amount the company has received in payment from tourists according to their countries of origin. Name the calculated field **IncomePerCountry**. The amounts must be formatted in currency value.

Example of the output (on the next page):

Country	IncomePerCountry
France	R 179 572.44
Germany	R 148 711.20
Spain	R 198 068.70
United Kingdom	R 159 085.56
United States of America	R 169 319.72

NOTE: The format of the **IncomePerCountry** field may differ from the example above.

(5)

1.7 **Menu Option G**

Display the destination, start date, seats available and the number of seats booked on all the tours that were NOT fully booked. A tour is NOT fully booked when the number of tourists who booked the tour is not equal to the number of seats available. Calculate and show the number of seats that were booked per tour. Name the calculated field **SeatsBooked**.

Example of the output for the first four records:

Destination	StartDate	Seats	SeatsBooked
Augrabies National Park	2012/03/17	15	10
Blyde River Canyon	2012/02/22	25	19
Cradle of Humankind	2012/03/12	18	3
Garden Route	2012/01/13	29	14

:

NOTE: The format of the date in the **StartDate** field may differ from the example above.

(7)

NOTE: If you want to test the menu options at this stage, use your backup copy of the **TourismDB** database.

- Enter your examination number as a comment in the first line of the file containing the SQL statements.
- Save your program.
- A printout for the code will be required.

[34]

QUESTION 2: OBJECT-ORIENTED PROGRAMMING

Blue Crane Tours offers different tour packages to various destinations around South Africa. The prices of the tours depend on the time of year the tours take place. As part of a marketing campaign, the company offers 10% discount on tours, depending on the number of tour members. Each client is also entered into a lucky draw and may be eligible for 25% discount based on the outcome of the lucky draw.

The files required for this question can be found in the folder named **Question2_XXXX** where XXXX refers to the programming language you have studied. You have been provided with a text file named **DataQ2.txt** and an incomplete program that consists of:

- A class unit (Delphi)/object class (Java) which describes the attributes of a tour and contains some methods
- A main form unit (Delphi)/test class (Java)

The text file contains data on an unknown number of tours offered by Blue Crane Tours. The details of each tour are stored using two lines of text in the file in the following format:

```
<name of the tour guide>&<destination of tour>  
<month during which the tour takes place> for <duration in days> days#<number of  
tourists taking the tour>
```

Example of the data of the first three tours in the text file **DataQ2.txt**:

```
Zane&Sun City Resort  
July for 3 days#10  
Jacky&Augrabies National Park  
March for 12 days#10  
Jacky&Garden Route  
January for 3 days#11
```

Do the following:

- Rename the folder for QUESTION 2 by replacing the name of the programming language you have studied with your examination number.
 - **Delphi programmers:**
 - Open the incomplete program file **Question2P.dpr**.
 - Add your examination number as a comment in the first line of both the class unit (**uQuest2**) and the main form unit (**Question2U**) files.
 - **Java programmers:**
 - Open the given incomplete object class **Quest2** and the test class **TestQuestion2**.
 - Add your examination number as a comment in the first line of both the object class (**Quest2**) and the test class (**TestQuestion2**).

- Compile and execute the program. The interface displays two menu options as indicated in the section labelled **QUESTION 2** in **ANNEXURE B (Delphi)/ANNEXURE C (Java)**.

2.1 **Do the following to complete the code in the class unit (Delphi)/object class (Java):**

The **uQuest2** unit (Delphi)/**Quest2** class (Java) contains the declaration of six attributes for a tour object and the get (accessor) methods for these attributes.

The six attributes are:

Description	Names of Attributes	
	Delphi	Java
Surname and initial of the tour guide	fGName	gName
Destination of the tour	fDName	dName
Month when the tour takes place	fMName	mName
Duration of the tour (number of days)	fNumD	numD
Number of tour members	fNumT	numT
Tariff per day	fTariff	tariff

Write code for additional methods as described below:

- 2.1.1 Write code for a method named **setTariff** with no parameters to calculate the tariff per day. The standard tariff per day depends on the month of the year when the tour takes place. The following applies:

Months	Tariff
April, September, December	R 1 250.00
January, February, August, October, November	R 1 000.00
March, May, June, July	R 900.00

(6)

- 2.1.2 Write code for a **constructor** method using parameter values to initialise the following attributes:

- Surname and initial of the tour guide
- Destination of the tour
- Month when the tour takes place
- Number of days (duration of the tour)
- Number of tour members

Use the **setTariff** method (written in QUESTION 2.1.1) to initialise the tariff attribute in the constructor.

(4)

- 2.1.3 Write code for a method called **shortenString** to shorten the name of a month by removing the vowels (a, e, i, o, u) from the name. The first character of the month name must not be removed even if it is a vowel. The shortened string should be returned.

Examples of some of the shortened strings:

August will be returned as **Agst**

September will be returned as **Sptmbr**

(5)

- 2.1.4 Write code for a method called **findLuckyChar** that will return an alphabetic character that is randomly selected from a destination's name. A space is not regarded as a lucky character. Both uppercase and lowercase characters are acceptable as lucky characters.

(5)

- 2.1.5 Write code for a **toString** method that will construct and return a string with labels and information about a tour object in the following format:

Month: <name of the month without vowels>
 Destination: <destination of the tour> with <name of tour guide> as the tour guide
 Price: <tariff per day in rand currency rounded to 2 decimal places> per day for a period of <number of days> days
 <number of tour members> tourists are taking this tour.

Example of the output when the string returned by the **toString** method is displayed:

Month: Octbr
Destination: Garden Route with Martin as the tour guide
Price: R 1 000.00 per day for a period of 11 days
20 tourists are taking this tour.

(5)

- 2.2 **Do the following to complete the code in the main form unit (Delphi)/test class (Java):**

- 2.2.1 Declare an array capable of storing 50 tour objects and a counter variable to keep track of the number of objects in the array.

(2)

- 2.2.2 Write code to test whether the text file exists.

If the text file exists, write code to read lines of text from the text file, extract the data, create a tour object and assign the object to the array.

If the text file doesn't exist, display a suitable message and terminate the program.

NOTE: The objects have to be assigned to the array before the menu options are displayed.

(12)

2.2.3 Write code to complete **menu option A** to do the following:

- Allow the user to enter the name of the month in which he/she wants to go on tour. Display a numbered list of the destinations of all the tours taking place during the month that was entered by the user.

Example of the output for the month of **February**:

Tours for the month of February	
=====	
Number	Destination
6	Richtersveld
32	Robben Island
33	Robben Island
35	Shakaland
37	Table Mountain National Park

NOTE: The numbers displayed must indicate the position of the tour in the array. (5)

- The user must then be prompted to choose one of the tours from the list by entering the number of the tour. The number entered by the user need not be validated.

Use the **toString** method to display all the information on the selected tour.

Example of the output when the user selects tour number 35 on the list:

Month: Fbrry
Destination: Shakaland with Abe as the tour guide
Price: R 1 000.00 per day for a period of 10 days
27 tourists are taking this tour.

(2)

- The user must then be prompted to choose a character from the name of the tour destination in order to stand a chance to be granted a 25% discount on the daily tariff of the tour. The name of the destination must be displayed as part of the message that prompts the user to enter a character.
- The program must use the **findLuckyChar** method to randomly select a 'lucky' character from the tour destinations. 25% discount on the daily tariff will be granted if the character entered by the user matches the character selected randomly by the **findLuckyChar** method.

If the discount is granted, a suitable message indicating the original tariff per day as well as the discounted tariff must be displayed.

If the discount is not granted, the lucky character selected by the program and a suitable message must be displayed. The original tariff per day must be part of the message.

Example of the output if the 25% discount was granted:

```
Congratulations! You have received 25%
discount on the daily tariff!
The tariff was R 1 000.00 per day. It has been
reduced to R 750.00 per day.
```

Example of the output if no discount was granted:

```
The lucky character was the letter d.
No discount. The tariff is still R 1 000.00
per day.
```

NOTE: The characters generated by your program may be different from those shown in the examples above due to the random selection of characters.

(8)

- Make sure that your examination number is entered as a comment in the first line of the class unit (Delphi)/object class (Java) as well as the main form unit (Delphi)/test class (Java).
- Save all the files.
- A printout of the code will be required.
- Print both the class unit (Delphi)/object class (Java) and the main form unit (Delphi)/test class (Java).

[54]

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

Blue Crane Tours needs statistics on the amount of foreign currency brought into the country by a number of tourists from various countries visiting South Africa. They also need to run a popularity test on some of the tours they offer.

The folder **Question3_XXXX** contains a text file named **DataQ3_XXXX.txt** where XXXX refers to the programming language you have studied.

The text file contains the declaration of an array named **arrData**. The array contains 40 strings with data on tourists who visited South Africa recently.

The data for each tourist in the array is in the following format:

`<Name of the tourist>@<country where the tourist is from>#<first two characters representing the tour destination captured as capital letters>#<total amount spent on the tour in the currency of the country of origin>`

The contents of the **arrData** array:

Rachel Delarosa@Canada#SH#11861,	Corradino Grande@Spain#RO#5788,
Lucas Herder@Germany#KR#7709,	Estotz Lizarazu@France#GA#12349,
Chynna Taylor@England#GA#8551,	Renata Di@Spain#RO#4906,
Ugs Boulot-Tolle@France#CA#7300,	Lena Bucholtz@Germany#GA#10344,
Maria Heimpel@Germany#SH#9438,	Julian Amstadter@Germany#RO#8840,
Sofie Mosbauer@Germany#GA#5894,	Fiona Green@England#CA#9094,
Sara Escobedo@Canada#KR#4381,	Nataly Mahan@Canada#RO#12642,
Wyatt Parham@Canada#SH#4799,	Noah Donovan@Canada#SH#3888,
Joseph Scott@England#SH#7928,	Emily Smith@England#KR#3110,
Adriana Mancuso@Spain#RO#3724,	Cassandra Wilder@Canada#KR#12583,
Tomasino Camporese@Spain#KR#6777,	Stacy Anderson@England#RO#3686,
Guiraud Bluteau@France#RO#11592,	Damian Friedman@Canada#RO#9012,
Anne Loef@Germany#KR#13035,	Terence Brown@England#SH#8180,
Lion Ghislieri@Spain#RO#14343,	Giraudetz Girardin@France#CA#11644,
Guglielmo Capriati@Spain#SH#5408,	David Geiberger@Germany#RO#9854,
Irisa Cooper@England#KR#11456,	Hayden Mcdonough@Canada#KR#7840,
Jonas Hipp@Germany#RO#3137,	Emily Kohler@Germany#GA#6509,
Emily Thul@Germany#RO#8551,	Gino Lazzaretti@Spain#CA#2329,
Alex Hofstater@Germany#GA#6751,	Peers Scott@England#RO#9470,
Liliana Horne@Canada#RO#14689,	Leon Kleinpaul@Germany#RO#15194

NOTE: The white space in the above text box is there only to facilitate reading.

The data captured in the first two strings can be interpreted as follows:

- Rachel Delarosa from Canada took a tour to Shakaland. She spent a total amount of 11 861 Canadian dollars.
- Corradino Grande from Spain took a tour to Robben Island. He spent a total amount of 5 788 euros.

The following five tour destinations are reflected in the data captured in the **arrData** array:

- Cape Winelands, represented as CA
- Garden Route, represented as GA
- Kruger National Park, represented as KR
- Robben Island, represented as RO
- Shakaland, represented as SH

Do the following:

- Rename the folder for QUESTION 3 by replacing the name of the programming language you have studied with your examination number.
- Create a new program/project/application.
- Enter your examination number as a comment in the first line of the program file you have created that will contain your code.

Java programmers:

If your solution consists of more than one class file, make sure to enter your examination number as a comment in ALL the class files.

- Save the program files using the question number as part of the filename in the renamed folder for QUESTION 3.
- Develop an interface as follows:
 - **Delphi programmers:**
 - Develop an interface to display a menu as indicated in the section labelled **QUESTION 3** in **ANNEXURE B**.
 - **Java programmers:**
 - Copy the code for displaying a menu as indicated in the section labelled **QUESTION 3** in **ANNEXURE C** from the **DataQ3_Java** text file.
- Copy the text for the declaration of the array which is supplied in the text file to your program file.
- Use the following code to complete the 'Quit'-menu option:
 - **Delphi programmers:**
 - `Application.Terminate;`
 - **Java programmers:**
 - `System.exit(0);`
- Complete the code for each menu option as follows (on the next page):

NOTE: You have to test your program by running the menu options in sequence, that is option A, then option B and finally option C.

3.1 Menu Option A

Data captured in the **arrData** array shows that tourists from France, Germany and Spain enter the country with their holiday money in euros as currency.

Write code to convert the total amount of euros the tourists from France, Germany and Spain brought into the country into South African rand. Use the exchange rate: 1 euro = R10,75. Display the total amount of holiday money in both currencies with suitable labels.

Example of the output:

Total amount in euro: 191 416
Total amount in South African rand: R 2 057 722.00

NOTE: The format of the **currency values** may differ from the example above.

(8)

3.2 Menu Option B

The touring company realised that many of the tourists visiting Robben Island are not English-speaking. They decided to divide the tourists taking tours to Robben Island into two groups: an English-speaking group and a non-English-speaking group.

Tourists from Canada and England are classified as English-speaking tourists.

The data in the **arrData** array needs to be adjusted so that it shows the two groups of tourists.

Write code to modify the data of tourists who visit Robben Island in the **arrData** array by changing the code for Robben Island (RO) in the array to ROEnglish or ROOther respectively.

Apply the following criteria:

- **ROEnglish** – for tourists from Canada and England
- **ROOther** – for tourists from France, Germany and Spain

Display a list of all the English-speaking tourists who took the tour to Robben Island.

Example of the output (on the next page):

```
List of English-speaking tourists to Robben Island
=====
Nataly Mahan
Stacy Anderson
Damian Friedman
Peers Scott
Liliana Horne
```

(11)

3.3 Menu Option C

The popularity of the tours needs to be determined. A star rating is used to evaluate the tours. The following criteria applies:

- One star represents a group of three tourists who took the tour.
- No star is awarded if fewer than three tourists took the tour.

Example:

- If five tourists take a tour, the rating of the tour will be one star.
- If six tourists take a tour, the rating of the tour will be two stars.

Display a list of all the tours, their star ratings and the actual number of tourists who took each tour (displayed between brackets).

HINT: Remember that the Robben Island tour was divided into two different tours, based on language (see QUESTION 3.2).

Example of the output:

```
Star rating of tours:
=====
Destination                Rating  Number of tourists
=====
Cape Winelands              *       ( 4)
Garden Route                **      ( 6)
Kruger National Park        **      ( 8)
Robben Island (English tour) *       ( 5)
Robben Island (Other tour)  ***     (10)
Shakaland                   **      ( 7)
```

(13)

- Make sure your examination number is entered as a comment in the first line of the program.
- Save all the files.
- A printout for the code will be required.

[32]

TOTAL: 120

ANNEXURE A: DATABASE STRUCTURE AND SAMPLE DATA

This annexure shows the database structure and sample data for the tables contained in the **TourismDB** database used in **QUESTION 1**.

tblTourists: This table contains data on tourists visiting South Africa who are taking tours arranged by Blue Crane Tours.

Table structure:

Field Name	Type	Size	Description
Number	AutoNumber	Long Integer	Number uniquely identifying each tourist
Surname	Text	20	Surname of the tourist
FirstName	Text	20	First name of the tourist
Gender	Text	1	Gender of the tourist (M for male and F for female)
Country	Text	25	Country of origin of the tourist
TourID	Number	Long Integer	ID of the tour that the tourist is taking
Deposit	Yes/No	True/False	Indicates whether a deposit has been paid
AmountPaid	Currency	Currency	Amount already paid

Sample data:

Number	Surname	FirstName	Gender	Country	TourID	Deposit	AmountPaid
1	Delarosa	Rachel	F	United States of America	42	False	R 0.00
2	Spina	Giacobbe	M	Spain	19	True	R 2 612.70
3	Grande	Corradino	M	Spain	11	False	R 0.00
4	Herder	Lucas	M	Germany	34	False	R 0.00
5	Ervin	Fiona	F	United States of America	22	False	R 0.00
6	Lizarazu	Estotz	M	France	24	True	R 2 600.28
7	Taylor	Chynna	F	United Kingdom	8	True	R 4 185.40
8	Di	Renata	F	Spain	4	True	R 5 006.40
9	Boulot-Tolle	Ugs	M	France	9	False	R 0.00
10	Bucholtz	Lena	F	Germany	44	True	R 3 910.68

tblTours: This table contains data on tours provided by Blue Crane Tours for the past two years.

Table structure:

Field Name	Type	Size	Description
TourID	Number	Long Integer	Number uniquely identifying the tour
Surname	Text	20	Surname of the tour guide
Initial	Text	2	Initial of the tour guide
Destination	Text	30	Destination of the tour
StartDate	Date/Time	yyyy/mm/dd	Date when the tour commences
EndDate	Date/Time	yyyy/mm/dd	Date when tour ends
Seats	Number	Integer	Maximum number of persons that can be accommodated
Price	Currency	Currency	Price for the tour per person

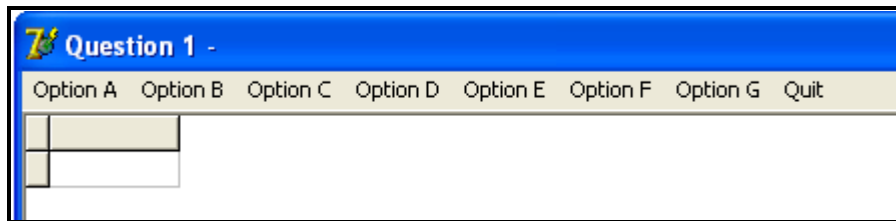
Sample data:

TourID	Surname	Initial	Destination	StartDate	EndDate	Seats	Price
1	Khumalo	J	Augrabies National Park	2011/03/19	2011/03/24	19	R 2 903.00
2	De Wet	H	Pilgrim's Rest	2011/04/24	2011/05/02	18	R 5 084.00
3	Nel	M	Victoria & Alfred Waterfront	2011/05/07	2011/05/16	24	R 5 480.00
4	Radebe	S	Kruger National Park	2011/08/06	2011/08/12	12	R 2 923.00
5	Botha	A	Drakensberg Park	2011/08/12	2011/08/20	29	R 4 071.00
6	Bandjies	J	West Coast National Park	2011/08/13	2011/08/23	13	R 2 520.00
7	Venter	M	Kruger National Park	2011/08/15	2011/08/25	16	R 4 945.00
8	Ramphopolos	Z	Robben Island	2011/09/18	2011/09/27	16	R 5 314.00
9	Smith	P	Cradle of Humankind	2011/09/28	2011/10/08	28	R 2 558.00
10	Roux	B	Richtersveld	2011/12/24	2012/01/12	30	R 5 184.00
11	Lemmer	G	Shakaland	2011/12/25	2012/01/16	20	R 4 937.00
12	Nel	M	Garden Route	2012/01/13	2012/01/18	29	R 4 828.00
13	Khumalo	J	Table Mountain National Park	2012/01/16	2012/01/19	11	R 4 924.00
14	Radebe	S	Table Mountain National Park	2012/02/16	2012/02/28	28	R 5 215.00
15	Bandjies	J	Robben Island	2012/02/16	2012/02/22	27	R 5 314.00

ANNEXURE B: DELPHI – GUI INTERFACES PER QUESTION

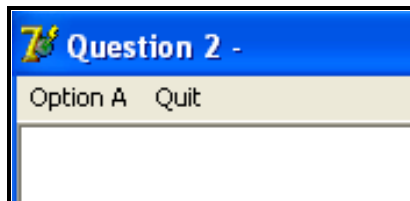
QUESTION 1

When you execute the program, the interface below will be displayed.



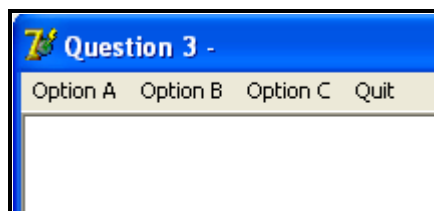
QUESTION 2

When you execute the program, the interface below will be displayed.



QUESTION 3

You are required to create the following interface as part of the solution for QUESTION 3. When you execute the program, the interface below must be displayed.



NOTE: Use the **MainMenu** component to create the menu.

ANNEXURE C: JAVA – GUI INTERFACES PER QUESTION**QUESTION 1**

When you execute the program, the interface below will be displayed.

```
MENU

Option A
Option B
Option C
Option D
Option E
Option F
Option G

Q - QUIT

Your choice?
```

QUESTION 2

When you execute the program, the interface below will be displayed.

```
MENU

Option A

Q - QUIT

Your choice?
```

QUESTION 3

Copy and use the code from the **DataQ3_Java** text file to create the following interface as part of the solution for QUESTION 3. When you execute the program, the interface below must be displayed.

```
MENU

Option A
Option B
Option C

Q - QUIT

Your choice?
```

ANNEXURE D: DELPHI – TROUBLESHOOTING DATABASE PROBLEMS

D.1 If you cannot use the database provided:

- Create your own database with the name **TourismDB** that includes a table named **tblTours** and another table named **tblTourists** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblTours.txt** and **tblTourists.txt**) to use as data for the different tables.
- The first line in the text files contains the field names to be used.

D.2 If your program cannot establish connectivity with the database:

- Make sure that the database **TourismDB** is in the same folder as your program for QUESTION 1. If this is not the case, copy the database file into the same folder as your program.

D.3 If your program establishes connectivity with the database but no data is displayed:

- Click on the ADOQuery component named **qryRec**.
- Click on the Ellipsis button (three dots) to the right of the 'ConnectionString' property in the Object Inspector.
- Click on the Build button, which takes you to the Data Link Properties dialogue box.
- Click on the Provider tab to open the Provider tab sheet and select Microsoft Jet 4.0 OLE DB Provider. Click on the Next button.
- The Connection tab sheet will be displayed. The first option on the Connection tab sheet provides an Ellipsis button (three dots) that allows you to browse and look for the **TourismDB** file. You will find this file in the QUESTION1 folder. Once you have found it, select the **TourismDB** file and then click on the Open button.
- Remove the user name Admin.
- Click on the Test Connection button.
- Click OK on each of the open dialogue windows.

ANNEXURE E: JAVA – TROUBLESHOOTING DATABASE PROBLEMS

E.1 If you cannot use the given database:

- Create your own database with the name **TourismDB** that includes a table named **tblTours** and a table named **tblTourists** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblTours.txt** and **tblTourists.txt**) to use as data for the different tables.
- The first line in the text files contains the field names to be used.

E.2 If your program cannot establish connectivity with the database:

- Make sure that the database **TourismDB** is in the same folder as your program for QUESTION 1. If this is not the case, copy the database file into the same folder as your program.

E.3 If you cannot establish connectivity with the database with the given files, use the following source code to ensure database connectivity:

```
try
{
    Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
    String filename = "TourismDB.mdb";
    String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=";
        database += filename.trim () + ";DriverID=22;READONLY=true}";
    Connection conn = DriverManager.getConnection (database, "", "");
}
catch (Exception e)
{
    System.out.println ("Unable to connect to the database");
}
```