

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2014

INFORMATION TECHNOLOGY P1

MARKS: 120

TIME: 3 hours



This question paper consists of 14 pages.

INSTRUCTIONS AND INFORMATION

1. This is a three-hour examination. 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. You require the files listed below in order to answer the questions. They are either on a CD issued to you, or the invigilator/educator will tell you where to find them on the hard drive of the workstation you are using or in a network folder.

QUESTION 1

Question1_u.pas

Question1_p.dpr

Question1_u.dfm

QUESTION 2

augp3.mdb

clsDriver.pas

Datam.dfm

Datam.pas

Q2_augrp.dfm

Q2_augrp.pas

Q2_augrp3unit.dfm

Q2_augrp3unit.pas

Q2_augrpunit.dfm

Q2_augrpunit.pas

Q2_australia.dpr

QUESTION 3

Question3_p.dpr

Question3_u.pas

Question3_u.dfm

abrivs.txt

3. If a CD containing the above files was issued to you, write your name and surname on the label.
4. Save your work at regular intervals as a precaution against power failures.
5. Rename the *P1 Data folder* as your *Name and Surname*.
6. Type in your name and surname as a comment in the first line of each program.
7. Read ALL the questions carefully. Do only what is required.
8. At the end of this examination session you will be required to hand in the CD with all the files with the work you have done or you must make sure that ALL the files with your work have been saved on the network as explained to you by the invigilator/educator.
9. Ensure that ALL files can be read.
10. During the examination you may use the HELP functions of the software. You may NOT make use of any other resource material.

SCENARIO

Formula One Racing or Grand Prix is a sporting event which takes place over three days (usually Friday to Sunday), with a series of practice and qualifying sessions prior to a race on Sunday. Booking for seats is usually done in advance using credit cards.

QUESTION 1: DELPHI PROGRAMMING

You have been asked to come up with a Boolean credit validation function using the Luhn algorithm to ensure that the credit card number is correct. The number of digits must be between 13 and 19.

The Luhn formula:

- Drop the last digit from the number. The last digit is what we want to check against
- Reverse the numbers
- Multiply the digits in odd positions (1, 3, 5, etc.) by 2 and subtract 9 from all the results greater than 9
- Add all the numbers together
- Mod by 10
- The number will be correct if your answer is the same as the last digit which was removed in the first step

Luhn example:

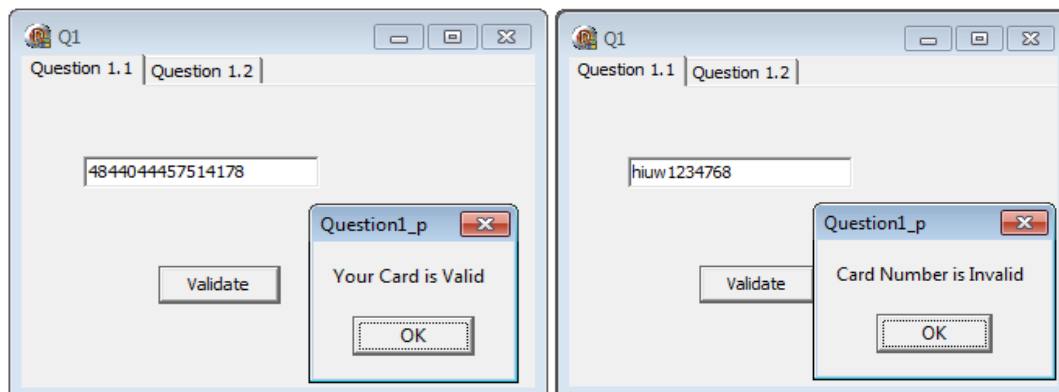
Step																	Total
Original number:	4	5	5	6	7	3	7	5	8	6	8	9	9	8	5	5	
Drop the last digit:	4	5	5	6	7	3	7	5	8	6	8	9	9	8	5		
Reverse the digits:	5	8	9	9	8	6	8	5	7	3	7	6	5	5	4		
Multiple odd digits by 2:	10	8	18	9	16	6	16	5	14	3	14	6	10	5	8		
Subtract 9 to numbers over 9:	1	8	9	9	7	6	7	5	5	3	5	6	1	5	8		
Add all numbers:	1	8	9	9	7	6	7	5	5	3	5	6	1	5	8		85
Mod 10:	85 modulo 10 = 5 (last digit of card)																

Open **Question1_dpr** and complete the questions:

1.1 **Question 1.1 Tab:**

FUNCTION:

- 1.1.1 Declare a Boolean function named CCValidate which receives a credit card number as a string variable and returns whether the credit card number is valid or invalid. (2)
- 1.1.2 Remove any letters from the credit card number. (5)
- 1.1.3 Check the length of the remaining number. The function must change to false if this criteria is not met. (3)
- 1.1.4
- Save last digit in a variable and remove it from remaining number
 - Start a loop to reverse the number
 - Multiply the numbers in odd position by 2; if the number is greater than 9 subtract 9 from that number
 - Add the numbers together
 - Mod the number by 10
 - Compare number with saved last digit
 - The function becomes false if this criteria is not met. (11)
- 1.1.5 **Validate Button:** Test the function by entering a credit card number. Display a suitable message whether the credit card number is valid or not. (4)



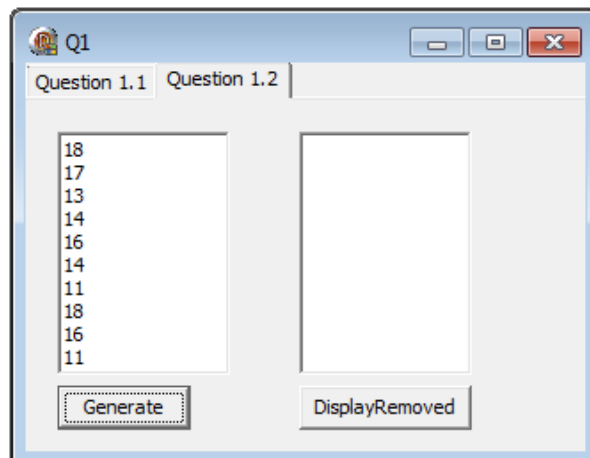
Test your program against these numbers: 4844044457514178, 30488409449900, 5415229692213231, hiuw1234768

1.2 Question 1.2 Tab:

When seats are booked, random ticket numbers are generated.

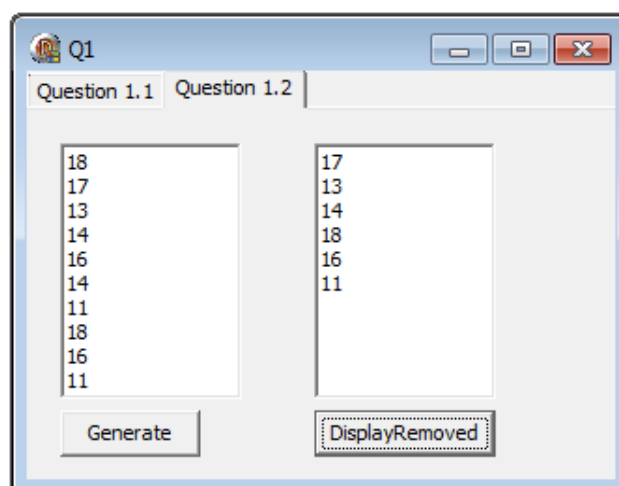
- 1.2.1 Declare two arrays with class scope, one to store the ten randomly generated numbers and the other to store the numbers without duplicates.

Generate button: generate 10 random numbers between 10 and 19 and display them in the richedit component.



(4)

- 1.2.2 **DisplayRemoved button:** remove any duplicate numbers and display the unique list in the richedit component.



- Enter your name and surname as a comment line in the first line of the file named Question1_u.pas.
- Save the program.
- Make a printout of the code of the Question1_u.pas file.

[35]

QUESTION 2: DELPHI PROGRAMMING

This question is intended to test object-orientated programming skills and database programming. You are required to produce a solution that includes all classes specified in the instructions. No marks will be allocated to any alternative solution such as one program not creating an object.

The database, *augp3.mdb*, supplied in the *Q2 Folder* contains results of the Formula 1 race in Australia in 2013. The database contains one table named *tblAUGP*.

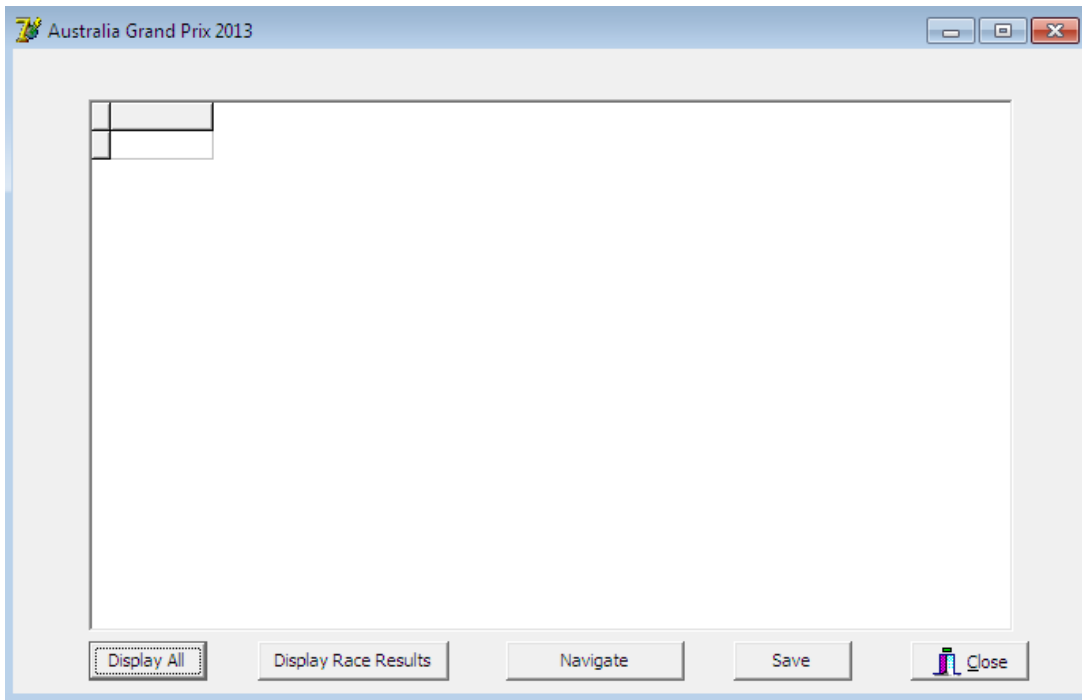
The *tblAUGP* table stores data on the registered drivers for 2013. The fields in this table are defined as follows:

Field Name	Type	Description
DRID	Text	Primary key for the table: Driver ID
Name	Text	Name and Surname of Driver
Country	Text	Nationality of Driver
Team	Text	Team of Driver
Time	Text	Time taken to complete the race

The following table is an example of the data contained in the table named *tblAUGP*:

DRID	Name	Country	Team	Time
FMBF02	Felipe Massa	Brazil	Ferrari	1:30'36.802
JBBM02	Jenson Button	Great Britain	McLaren/Mercedes	1:31'24.855
JVFT01	Jean-ric Vergne	France	Toro Rosso/Ferrari	1:31'27.082

Load *Q2_australia.dpr* and you should be presented with the following multiform project.



Connectivity to the database is provided for you through the Data Module, AUDatam.

If the connection to the database is giving you problems ensure that the data aware components on the AUDatam form, have the following properties:

- The **ADONauggp3.ConnectionString** property: Click on the two dots and select the Microsoft Jet 4.0 OLE DB Provider driver. Under the file name put AUGP3.mdb. Remove the user name and select blank password.
- The **ADONauggp3.LoginPrompt** property should be false.
- The **ADONauggp3.Connection** property should be ADONauggp3.
- The **ADONauggp3.Tablename** property should be tblAUDB.
- The **DataNauggp3.Dataset** property should be ADONauggp3.

2.1 Open the unit called *clsDriver*.

2.1.1 Define a class with the name *TDriver*. (1)

2.1.2 The class must contain the following private fields with appropriate data types:

fName: to hold the name and surname of driver

fCountry: to hold the name of the driver's country

fTeam: to hold the name of the driver's team

fTime: to hold the time taken by the driver in the format: h:min'sec (4)

2.1.3 Create a constructor named *Create* that will receive four parameters and initialise all the fields. (7)

2.1.4 Create a method *getfName* that will return the name (not surname) of the driver. (3)

- 2.1.5 Create a method `getAverage` that will return the average speed of each driver in the format as shown in the example.

Example:

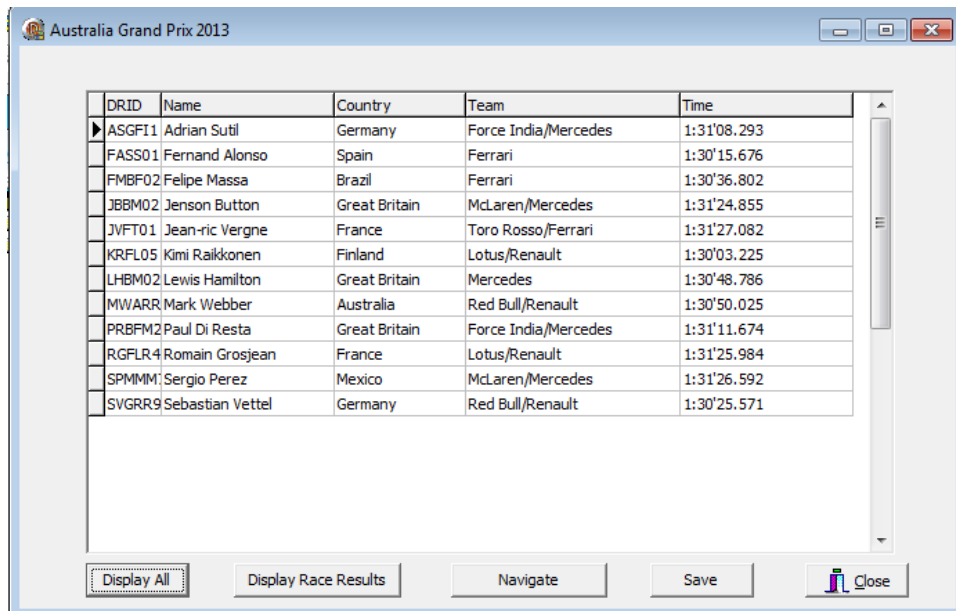
The Average speed of Fernand Alonso is 63.72 m/s

The average speed is calculated by multiplying 58 laps and 5 900 m then divide by the time taken in seconds.

Time in seconds = (hoursx60X60+minutesx60+seconds) (5)

- 2.1.6 Create a method `toString` that will return the name, country, team and time (h:min:sec) as one string. (6)

- 2.2 Write code for the *Display All* button on the form *frmAustralia* to display all the contents of the database table. Do NOT use the class designed in QUESTION 2.1 to achieve this.

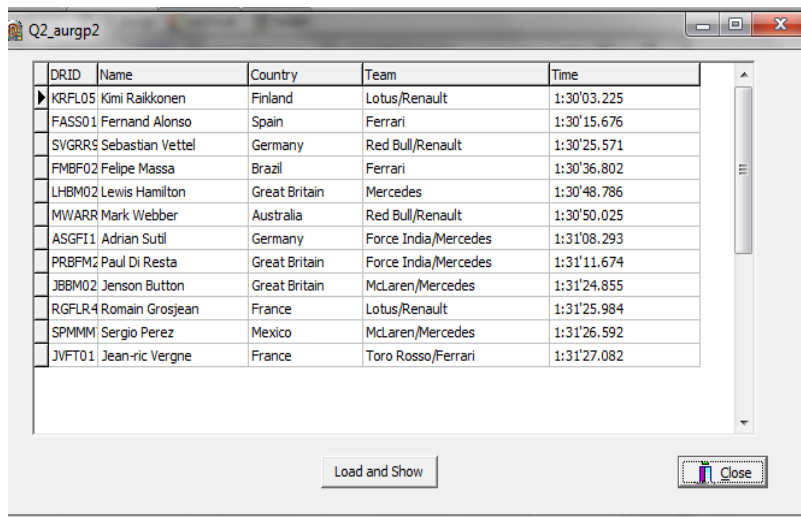


(2)

2.3 With the button Display Race Results:

2.3.1 Write code on the btnResults onClick event handler to show the Q2_aurgp2 form.

On the OnActivate event handler of the form, add code to display the results on the Stringgrid component, arranged from the first driver position to the last. The driver with the least lap time is the winner. Do NOT use the class designed in QUESTION 2.1 to achieve this.



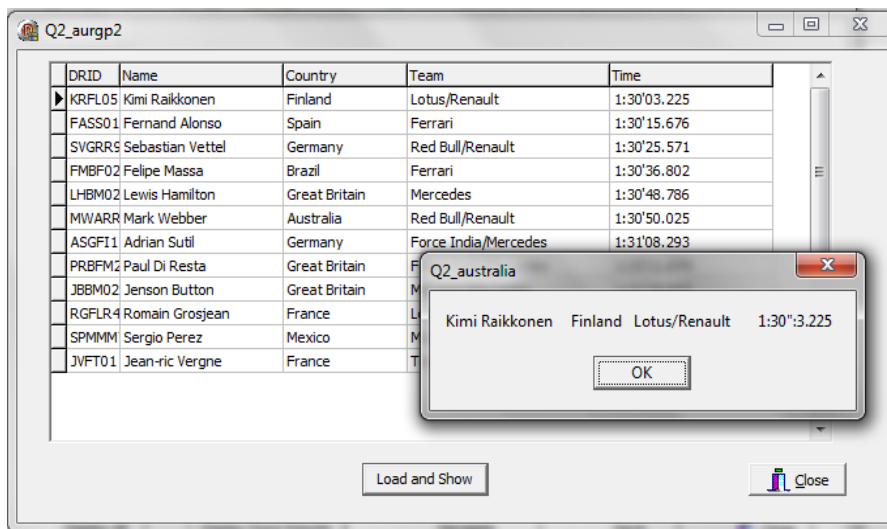
The screenshot shows a form titled "Q2_aurgp2" with a table displaying race results. The table has five columns: DRID, Name, Country, Team, and Time. The data is as follows:

DRID	Name	Country	Team	Time
KRFL05	Kimi Raikkonen	Finland	Lotus/Renault	1:30'03.225
FASS01	Fernand Alonso	Spain	Ferrari	1:30'15.676
SVGRR5	Sebastian Vettel	Germany	Red Bull/Renault	1:30'25.571
FMBF02	Felipe Massa	Brazil	Ferrari	1:30'36.802
LHBM02	Lewis Hamilton	Great Britain	Mercedes	1:30'48.786
MWARR	Mark Webber	Australia	Red Bull/Renault	1:30'50.025
ASGF11	Adrian Sutil	Germany	Force India/Mercedes	1:31'08.293
PRBFM2	Paul Di Resta	Great Britain	Force India/Mercedes	1:31'11.674
JBBM02	Jenson Button	Great Britain	McLaren/Mercedes	1:31'24.855
RGFLR4	Romain Grosjean	France	Lotus/Renault	1:31'25.984
SPMMM	Sergio Perez	Mexico	McLaren/Mercedes	1:31'26.592
JVFT01	Jean-ric Vergne	France	Toro Rosso/Ferrari	1:31'27.082

Below the table is a "Load and Show" button and a "Close" button.

(4)

2.3.2 On the form Q2_aurgp2 write code on the btnWinner to display the winner of the race using a ShowMessage component. Use the class designed in QUESTION 2.1 to display the information.



The screenshot shows the same "Q2_aurgp2" form as before, but with a message box titled "Q2_australia" overlaid. The message box displays the winner's information: "Kimi Raikkonen Finland Lotus/Renault 1:30':3.225". There is an "OK" button in the message box.

(10)

2.4 With the button Navigate:

2.4.1 Write code on the btnNavigate to show the form Q2_aurpg3 (2)

2.4.1 On the Q2_aurpg form, do the following:
Place the relevant components on the form as shown below to allow you to navigate through the form, and write any code if necessary.

The screenshot shows a form titled 'Q2_aurpg3' with the following fields and values:

- DRID: ASGF11
- Name: Adrian Sutil
- Country: Germany
- Team: Force India/Mercedes
- Time: 1:31'08.293
- Average Speed: (empty)

At the bottom, there is a navigation bar with buttons: back, forward, home, search, and a 'Close' button.

(3)

2.4.3 Insert code on the navigation button which will allow you to display the average speed on a label simultaneously. **Use the class you have defined in QUESTION 2.1 to accomplish this.**

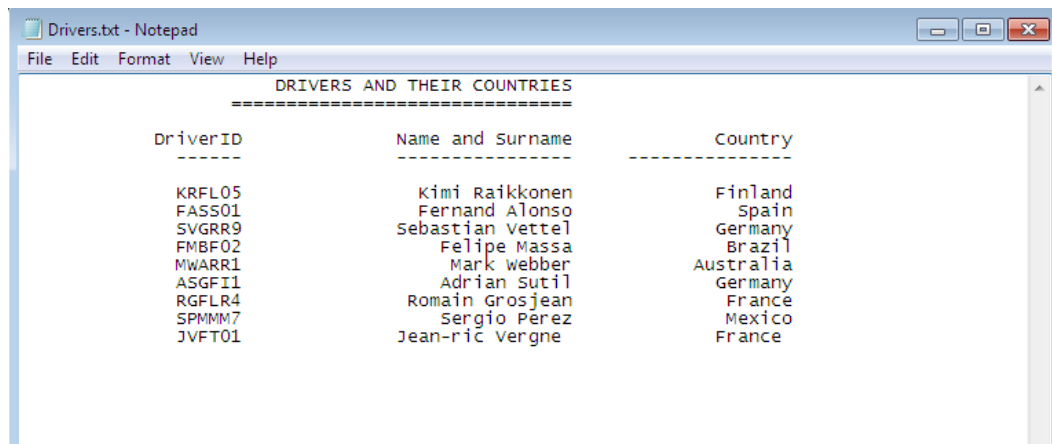
The screenshot shows the same form as before, but with the following changes:

- DRID: FASS01
- Name: Fernand Alonso
- Country: Spain
- Team: Ferrari
- Time: 1:30'15.676
- Average Speed: The Average speed of Fernand Alonso is 63.72 m/s

The navigation bar at the bottom now includes an additional button (likely 'Calculate') and the 'Close' button.

(11)

- 2.5 Write code for the btnSave buttonclick event handler to allow you to add/write the Driver ID, Name, and Nationality of each driver who is not from Great Britain in a file called **Drivers.txt**. Below is a screen shot of the text file. Put appropriate headings and define the columns.



DRIVERS AND THEIR COUNTRIES		
DriverID	Name and Surname	Country
KRFL05	Kimi Raikkonen	Finland
FASS01	Fernand Alonso	Spain
SVGRR9	Sebastian Vettel	Germany
FMBF02	Felipe Massa	Brazil
MWARR1	Mark webber	Australia
ASGF11	Adrian Sutil	Germany
RGFLR4	Romain Grosjean	France
SPMM7	Sergio Perez	Mexico
JVFT01	Jean-ric Vergne	France

(16)

- Enter your name and surname as a comment line in the first line of the file names clsdriver.pas, Q2_aurgp.pas, Q2_aurgp2.pas, Q2_aurgp3.pas.
- Save the program.
- Make a printout of the codes of the files clsdriver.pas, Q2_aurgp.pas, Q2_aurgp2.pas, Q2_aurgp3.pas.

[74]

QUESTION 3: DELPHI PROGRAMMING

The Formula 1 Grand Prix Driver's and Constructor's Championships are decided by points, which are awarded according to the place in which a driver finishes at each grand prix. The scoring system from 2013 is:

Driver completed 90% of winner's race distance	
1st place	25 points
2nd place	18 points
3rd place	15 points
4th place	12 points
5th place	10 points
6th place	8 points
7th place	6 points
8th place	4 points
9th place	2 points
10th place	1 point
11th place onwards	No points

Only the first, second and third placed drivers for a particular race get to be on the podium. **We will analyse only the first ten (10) Grand Prix Circuit results of 2013 and the top ten (10) positions.**

Note the following:

- *First three lines of the file **abrivs.txt** are in this format:*

<abbreviation>,<Surname and First Name>

VET,Sebastian Vettel

WEB,Mark Webber

ALO,Fernando Alonso

- A two-dimensional array of Driver's Abbreviations with 10 rows and 10 columns called *arrResults*, is used to represent the results of 10 Grand Prix Circuits
- A one dimensional array of Circuit Names called *arrCircuits* contains the names of the 10 Circuits of 2013

3.1 Menu Option: Load

This menu option is used to load the data from the text file **abrivs.txt** into a two dimensional *arrAbrevs* (which has already been declared for you). Row 1 will store the abbreviations and row 2 will store the full name of the driver.

(10)

3.2 Menu Option: Display

Display the information stored in *arrAbrevs* in the richedit component provided in the following format: abbreviation<space>name. (Eg. WEB Mark Webber).

Display the information stored in *arrPositions* in the stringgrid component as indicated in the diagram.

Add the positions in the first row and the circuits in the first column of the StringGrid as indicated below:

2013 Formula 1 Grand Prix Result Analysis After 10 Circuits

	1	2	3	4	5	6	7	8	9	10
Rolex Australian Grand Prix	VET	WEB	HAM	MAS	ALO	ROS	RAI	GRO	DIR	BUT
Petronas Malaysia Grand Prix	VET	MAS	ALO	HAM	WEB	ROS	BUT	SUT	PER	RAI
UBS Chinese Grand Prix	HAM	RAI	ALO	ROS	MAS	GRO	RIC	BUT	VET	HUL
Gulf Air Bahrain Grand Prix	ROS	VET	ALO	MAS	DIR	SUT	WEB	RAI	HAM	BUT
57° Gran Premio de España	ROS	HAM	VET	RAI	ALO	GRO	WEB	PER	MAS	DIR
71e Grand Prix de Monaco	ROS	HAM	VET	WEB	RAI	ALO	PER	SUT	BUT	VER
Grand Prix du Canada	VET	HAM	BOT	ROS	WEB	ALO	VER	SUT	HUL	RAI
Santander British Grand Prix	HAM	ROS	VET	WEB	RIC	SUT	GRO	RAI	ALO	BUT
Grosser Preis Santander von Deutschland	HAM	VET	WEB	RAI	GRO	RIC	MAS	ALO	BUT	HUL
Magyar Nagydíj	HAM	VET	GRO	ROS	ALO	RAI	MAS	RIC	PER	WEB

Driver List:

- WEB Mark Webber
- ALO Fernando Alonso
- MAS Felipe Massa
- BUT Jenson Button
- PER Sergio Pérez
- RAI Kimi Räikkönen
- GRO Romain Grosjean
- ROS Nico Rosberg
- HAM Lewis Hamilton
- HUL Nico Hülkenberg
- GUT Esteban Gutierrez
- DIR Paul Di Resta
- SUT Adrian Sutil
- CAL James Calado
- MAL Pastor Maldonado
- BOT Valtteri Bottas
- VER Jean-Éric Vergne
- RIC Daniel Ricciardo
- PIC Charles Pic
- VDG Giedo van der Garde
- KOV Heikki Kovalainen
- BIA Jules Bianchi
- GON Rodolfo Gonzalez
- CHI Max Chilton

(8)

3.3 Menu Option: Podium

Prompt the user to enter the abbreviation of any driver and then determine the number of times the driver has been on the winner's podium. To have been on the podium the driver must be amongst the top 3 drivers during a race.

Driver Podium Search

Enter Abbreviations of Driver here

Vet

OK Cancel

results

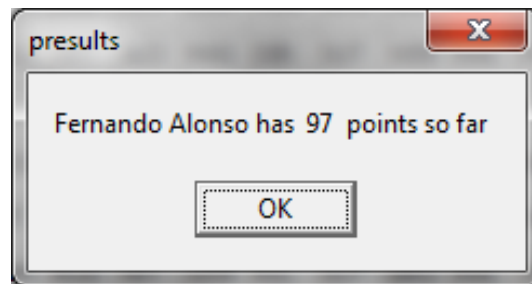
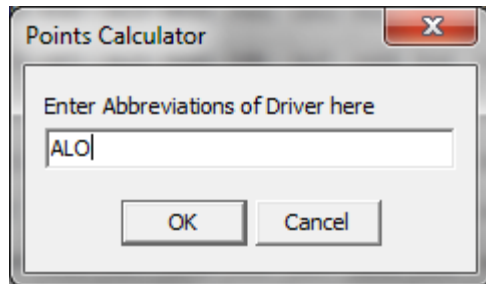
Sebastian Vettel has been on the podium 9 times so far

OK

(13)

3.4 Menu Option: Points

Prompt the user to enter the abbreviation of any driver and then determine the points of the driver according to the 2013 scoring system mentioned above.



(10)

- Enter your name and surname as a comment line in the first line of the file named Question3_u.pas.
- Save the program.
- Make a printout of the code of the Question3_u.pas file.

[41]

TOTAL: 150