basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

KWAZULU – NATAL

GRADE 11

INFORMATION TECHNOLOGY P1

JUNE 2018

MARKS: 150

TIME: 3 hours

This question paper consists of 11 pages.

INSTRUCTIONS AND INFORMATION

1. This paper consists of THREE questions. Candidates must answer ALL THREE questions.

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.

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 search, sort and selection, must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.

8. All data structures must be defined 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 full name, grade and division appears as a comment in every program that you code, as well as on every event indicated.

The following files are in you examination folder

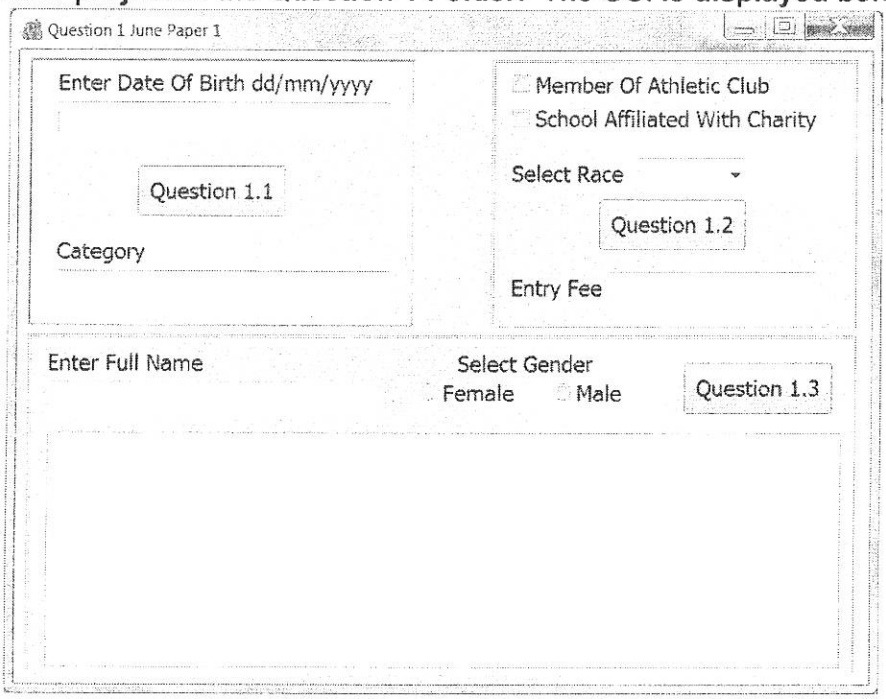
Question1	
frmQuestion1.dcu frmQuestion1.dfm frmQuestion1.pas	Question1.dpr Question1.dproj Question1.res
Question2	
frmQuestion2.dcu frmQuestion2.dfm frmQuestion2.pas Registered.txt	Question2.dpr Question2.dproj Question2.res
Question3	
frmQuestion3.dcu frmQuestion3.dfm frmQuestion3.pas Registered.txt	Question3.dpr Question3.dproj Question3.res

SCENARIO

Winter Is Coming is a charitable organisation that is holding a Fun Run/Walk/Marathon to raise funds to assist underprivileged people this winter. In order to facilitate the smooth running of this event, the organisers have asked you to assist with certain organisational aspects to ensure the integrity and correctness of records, since the previous individual, Daen. E. Rys, has left the organisation. Before leaving, however, she did start designing the software. You are required to complete what she began.

QUESTION 1

Open the Question1 project in the Question 1 Folder. The GUI is displayed below.

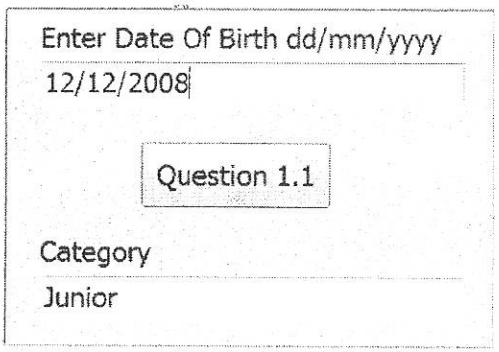


You are required to write code for each of the following:

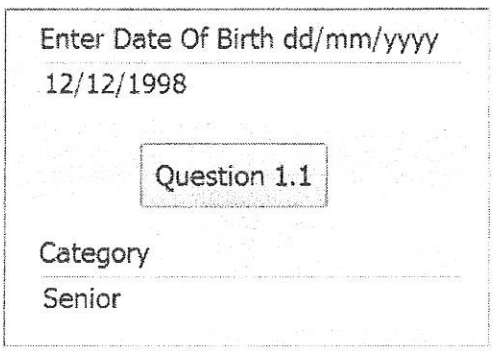
1.1 **Button Question 1.1**

Extract the date of birth entered by the user in the input component and determine and output (in the output component provided) whether the entrant is a **Junior** (16 years or younger) or a **Senior**.

Sample Run 1:



Sample Run 2:



(8)

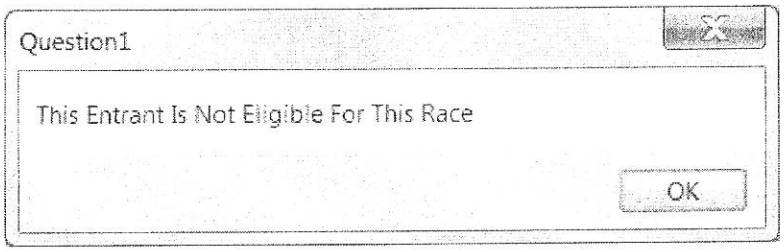
1.2 **Button Question 1.2**
Write code to determine the **EntryFee** that is payable. The **EntryFee** is calculated as follows:

Junior	R45.00	
Senior	R75.00	
Member Of Athletic Club	8% Discount	Use the Checkboxes to determine this
School Affiliated With charity	2.5% Discount	

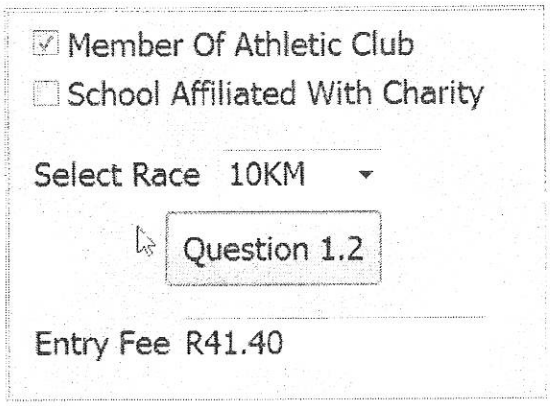
Furthermore, a Junior is ***only eligible to participate in either the 5KM or 10KM*** race and a Senior is ***only eligible for the 20KM or 50KM*** race(***Use the Category determined in Question 1.1***). If an incorrect race is selected (according to **Category**), an error message must be displayed and the focus set back to the first element in **Race** combo box.

If the correct race has been selected, determine and output the **EntryFee** payable (as **Currency**) in the display component provided.

Sample Run : If the 20KM Race was selected for a Junior



Sample Run : If the 10KM Race was selected for a Junior



(12)

1.3 **Button Question 1.3**

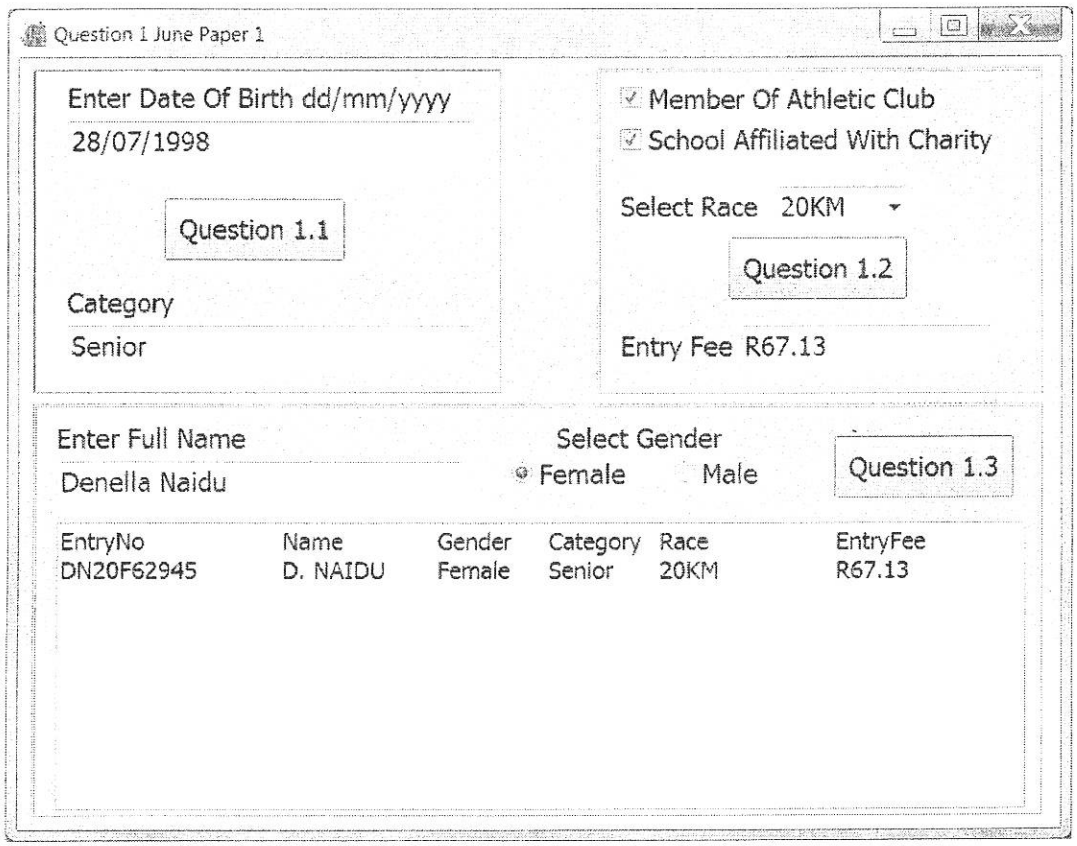
Write code to allow the user to enter his/her **Full Name** (***Assume there is only ONE Firstname and ONE Surname***) and select his/her **Gender**. Thereafter generate an Entry Number (EntryNo) for the entrant by joining:

- The uppercase initial of each name in the Full Name
- The age of the entrant (from Question 1.1)
- F for Female or M for Male
- The number of vowels in the Full Name
- A random number between 1000 to 5000(inclusive)

For each correct entry captured, the following details must be displayed in the output area provided. The output format

EntryNo Initial & Surname Gender Category Race Entry Fee

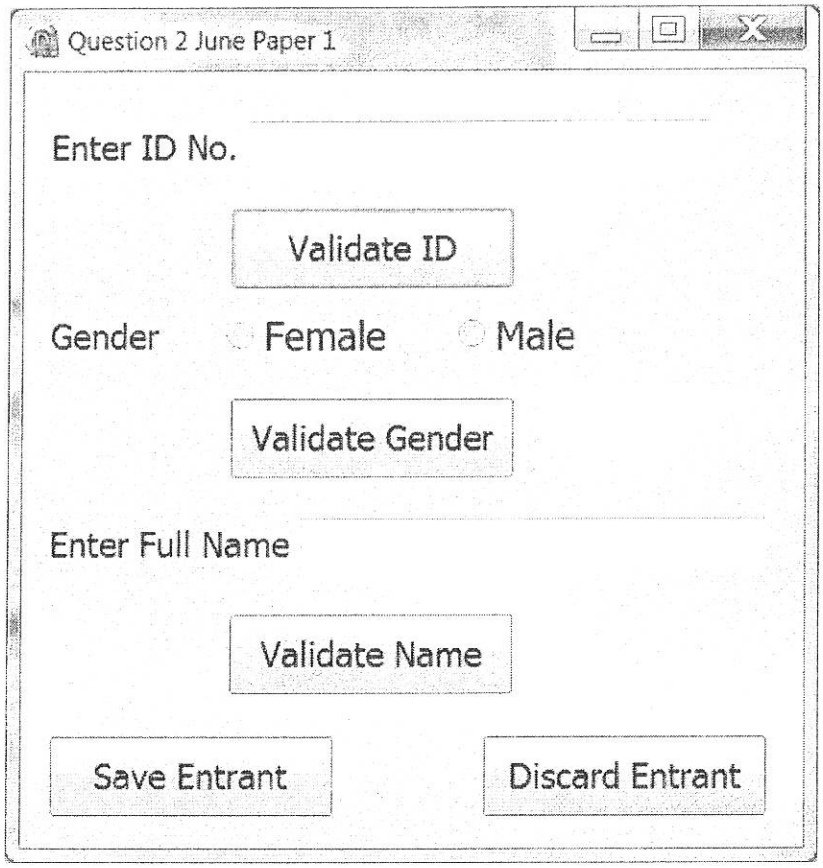
Sample Input/Output



(15)
[35]

QUESTION 2

Open the Question2 project in the Question 2 Folder. The GUI is displayed below.



2.1 Write code for the following methods:

2.1.1 Write a Function **isValidID** that will receive an ID Number as a String and Return TRUE if the ID Number is valid, otherwise it should return FALSE.

An ID Number consists of 13 Digits. The 13th digit is the checksum digit used to verify whether the ID Number is valid or not.

An ID Number is validated as follows:

Sample Input : 9807280044081

Position	1	2	3	4	5	6	7	8	9	10	11	12	***13***
Digit	9	8	0	7	2	8	0	0	4	4	0	8	***1***

Work with the first 12 digits:

1. Determine the sum of all digits stored in an ODD position

   9 + 0 + 2 + 0 + 4 + 0 = 15

2. Join the digits stored at EVEN positions, convert to Integer and multiply by 2

   878048 × 2 = 1756096

3. Determine the sum of the digits of the answer in 2.

   1 + 7 + 5 + 6 + 0 + 9 + 6 = 34

4. Add the answers from 1 and 3

   15 + 34 = 49

5. From the answer in 4, subtract the second digit from 10. If this matches the 13th digit in the ID Number, then the ID Number is valid.

   10 – 9 = 1 → This matches the 13th digit, therefore the ID Number is valid.

(16)

2.1.2 Write a Function **isValidGender** that will receive the ID Number and Gender as Strings and return 1 if the Gender matches the ID Number, otherwise it should return 0.

In an ID Number, the 7th, 8th, 9th and 10th digits together represent the Gender of an individual.

Example : If the ID Number received was

9 8 0 7 2 8 0 0 4 4 0 8 1

These four digits represent the gender
If these digits together as a single number is less than 5000, then the Gender is FEMALE, otherwise it is MALE.

Compare the Gender represented by this number with the Gender received to determine the Function return value

(8)

2.1.3 Write a Function **isValidName** that will receive a Full Name and return TRUE if the Full Name is valid, otherwise it should return FALSE.

A Valid Full Name can contain the following characters ONLY:

- Letters of the Alphabet ('a'..'z' OR 'A'..'Z')
- Spaces
- Hyphens (–)

(8)

2.1.4 Write a Procedure **resetForm** that wlll

- Disable all buttons excluding the Validate ID button
- Clear all inputs and selections
- Send the cursor back to the ID Number input component

(3)

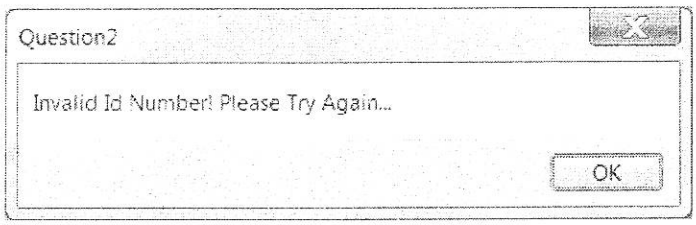
2.2 The only button that is enabled is the Validate ID button.
Write code to complete the functionality of each of the following buttons:

2.2.1 **Button Validate ID**

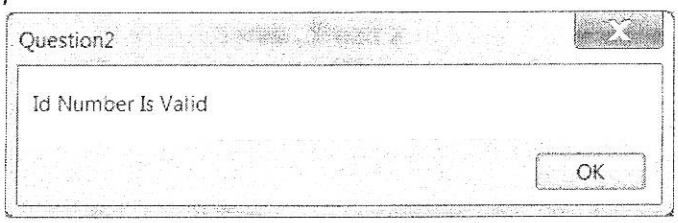
Get the ID Number input by the user and use the isValidID function to validate the ID Number. If it is valid, enable the Validate Gender button, otherwise

- Display and error message asking the user to re – enter ID Number
- Clear the ID Number input component
- Send the cursor back to the ID Number input component

Sample Input : 9812040314088



Sample Input : 9812040314089



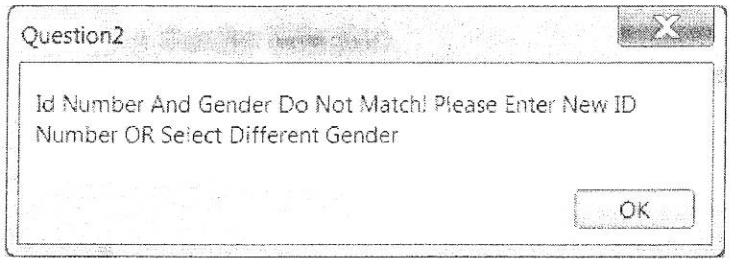
(4)

2.2.2 **Button Validate Gender**

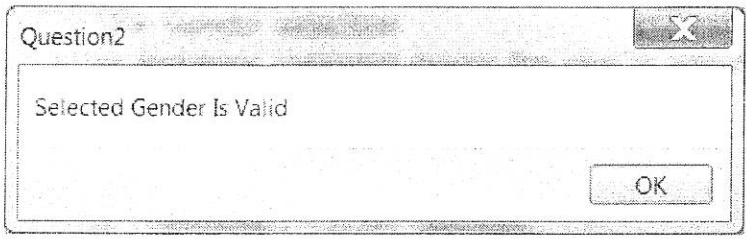
Get the ID Number input and the Gender selected and use the isValidGender Function to validate the Gender. If the Gender is valid, enable the Validate Name button, otherwise

- Clear the Gender selection
- Display and error message asking the user to select Gender again

Sample Input : 9812040314089
Gender Selected is Male



Sample Input : 9812040314089
Gender Selected is Female



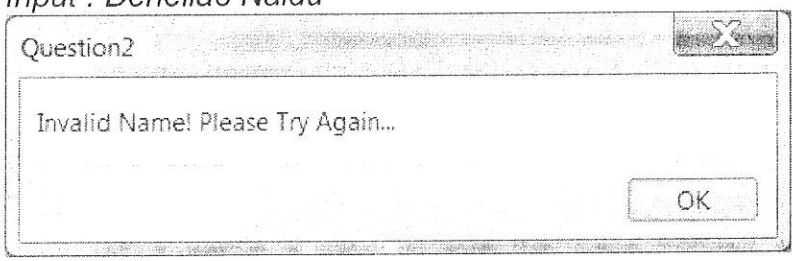
(9)

2.2.3 Button Validate Name

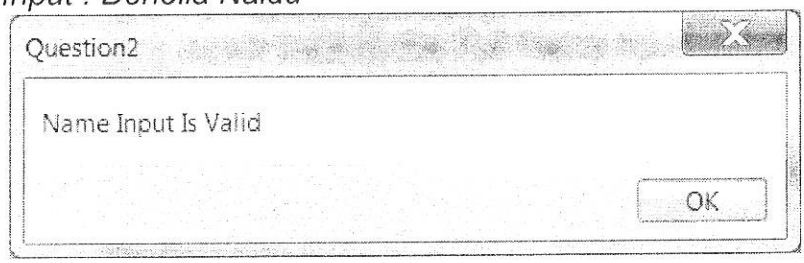
Get the Full Name input and use the isValidName function to validate the Full name. If the Full Name is valid, enable the Save Entrant and Discard Entrant buttons, otherwise

- Display and error message asking the user to re – enter Full Name
- Clear the Full Name input component
- Send the cursor back to the Full Name input component

Sample Input : Denella8 Naidu



Sample Input : Denella Naidu



(5)

2.2.4 Button Save Entrant

Add the ID Number, Full Name and Gender to the existing file **Registered.txt** in the following format:

IDNo#FullName#Gender

Example of saved data:
9807280044081#Denella Naidu#Female

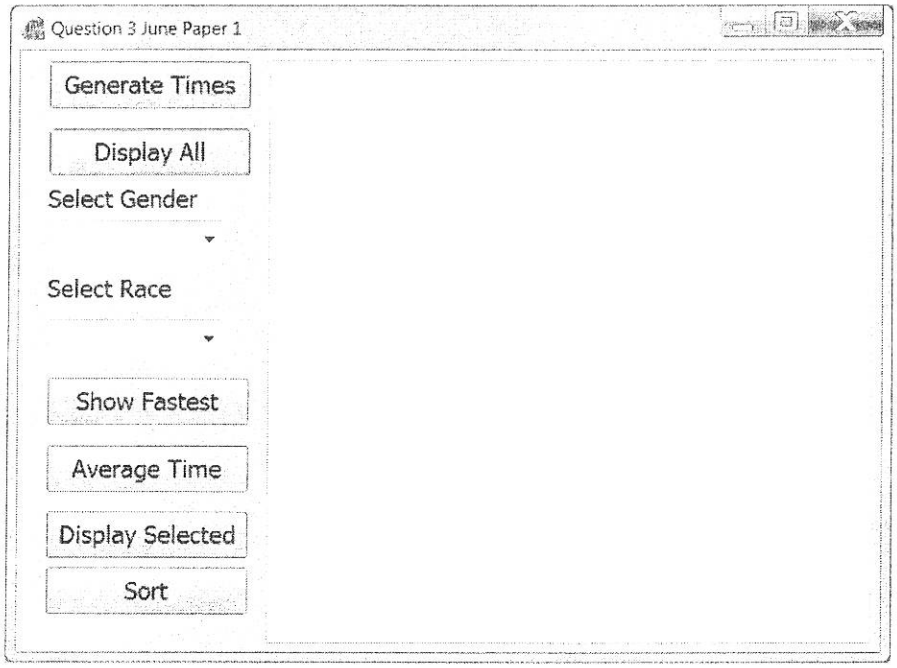
Use the resetForm procedure to reset the form for a new entrant's details. (8)

2.2.5 Button Discard Entrant

Use the resetForm procedure to reset the form for a new entrant's details. (1)
[62]

QUESTION 3

Open the Question3 project in the Question 3 Folder. The GUI is displayed below.



Three arrays ***gender***, ***race*** and ***time*** have been declared globally. These arrays are used to store the ***Gender(M or F)***, the ***Race(in KM)*** registered for and each entrant's ***time***(Finishing Time in minutes). A global variable ***numEntrants*** has also been declared which represents the total number of elements in each array.

Sample contents of the arrays are :

gender
F
M
F
F
M

race
5
20
50
5
10

The time array is empty

Write code to complete the functionality for each of the following buttons:

3.1 Button Generate Times

Finishing times for each entrant must be randomly generated according to the following criteria:

If race is 5KM	Generate a value from 20 to 60
If race is 10KM	Generate a value from 61 to 90
If race is 20KM	Generate a value from 250 to 350
If race is 50KM	Generate a value from 390 to 500

Sample values that could be generated :

gender
F
M
F
F
M

race
5
20
50
5
10

time
45
280
412
58
83

Since the times are randomly generated, your generated values may differ from the sample above. (6)

3.2 **Button Display All**
Add code to display the contents of all the arrays in tabular form with suitable headings.

NOTE : The finishing time for each entrants must be converted to Hours and Minutes which must be displayed

Sample Display

Gender	Race	Finishing Time
M	20 KM	4:34
F	20 KM	5:10
F	10 KM	1:1
F	10 KM	1:16
M	50 KM	7:6
M	5 KM	0:47
M	5 KM	0:31
F	10 KM	1:26
M	5 KM	0:44
M	10 KM	1:25
F	20 KM	5:0
M	10 KM	1:25

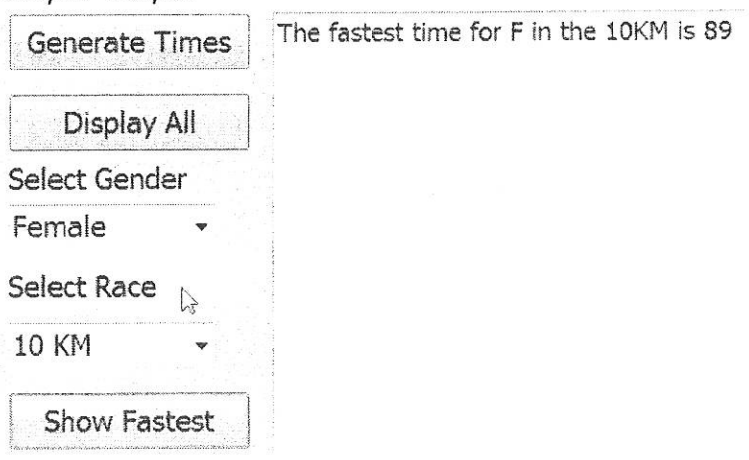
(8)

3.3 The code you must complete for the following buttons is dependent on the user selection of Gender and Race from the combo boxes.

3.3.1 **Button Show Fastest**

Write code to determine and display the fastest time for the selected Gender in the selected Race.

Sample Output

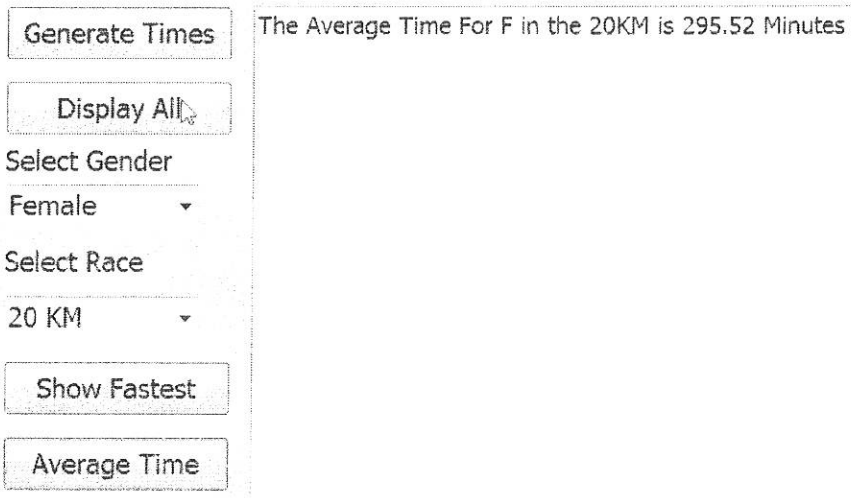


(16)

3.3.2 Button Average Time

Write code to determine and display the average time(rounded off to 2 decimal places) for the selected Gender in the selected Race.

Sample Output



(8)

3.3.3 Button Display Selected

Write code to display **ALL** entrants for the selected Gender in the selected Race.

Sample Output

Gender	Race	Finishing Time
F	10 KM	1:1
F	10 KM	1:16
F	10 KM	1:26
F	10 KM	1:8
F	10 KM	1:7
F	10 KM	1:26
F	10 KM	1:23
F	10 KM	1:9
F	10 KM	1:4
F	10 KM	1:29
F	10 KM	1:5
F	10 KM	1:22
F	10 KM	1:12

(4)

3.4 **Button Sort**

Write code to sort the arrays in from fastest to slowest times, grouped according to race.

Sample Output after Sort

Gender	Race	Finishing Time
M	5 KM	0:21
M	5 KM	0:22
M	5 KM	0:23
F	5 KM	0:24
F	5 KM	0:24
M	5 KM	0:25
M	5 KM	0:26
F	5 KM	0:27
F	5 KM	0:27
M	5 KM	0:28
F	5 KM	0:28

(11)
[53]

GRAND TOTAL : 150