

**PROTEA HEIGHTS ACADEMY**

**SEPTEMBER 2023**

**INFORMATION TECHNOLOGY**

**PAPER 1 - PRACTICAL**

**This paper consists out of 5 pages.**

Examiner: MF Moderator: PL

TIME: 1 HOUR MARKS: 55

**INSTRUCTIONS AND INFORMATION:**

|  |  |
| --- | --- |
|  |  |
| 1. | This question paper consists of two questions. You must answer BOTH QUESTIONS. |
|  |  |
| 2. | The duration of this examination is **1 hour**. 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. | Make sure that you answer the questions according to the specifications that are given in each question. Marks will only be awarded according to the set of requirements. |
|  |  |
| 4. | Answer only what is asked for in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation. |
|  |  |
| 5. | 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. |
|  |  |
| 6. | **Routines such as search, sort and selection** must be developed from first principles. You may not use the built-in features of the programming language for any of these routines. |
|  |  |
| 7. | All data structures must be defined by you, the programmer, unless the data structures are supplied. |
|  |  |
| 8. | You must **save your work regularly** on the disk space allocated to you for the examination OR the flash you have been given. |
|  |  |
| 9. | Make sure that **your** **Name and Surname** appears as a comment in every program that you code. |
|  |  |
| 10. | At the end of the examination session, you must make sure that all your work has been saved on the disk space allocated to you for this examination session OR you must hand in the flash with all your work saved on it. **Ensure that all the** **files can be read**. |
|  |  |
| 11. | The files you need to complete this question paper have been given to you on the disk space allocated to you in the form of a compressed file: **IT11 Data files.** |
|  |  |
|  | Once extracted, the following list of files will be available in the folder:  Delphi files: |
|  |  |
|  |  |
|  | **Change the name** of the **IT11 Data files** folder **to your name and surname**. |

|  |  |  |
| --- | --- | --- |
| **QUESTION 1: DATABASE MANIPULATION** | | |
|  |  |  |
| This section consists of question 1.1. to 1.5. The following important notes are applicable to all questions:   * You are NOT allowed to modify or add to the supplied data in any way. * Good programming techniques must be applied when coding your solutions. * NO marks will be assigned for hardcoding. Use control structures and variables where necessary. | | |
|  |  |  |
| **Scenario:** | | |
|  |  |  |
| Your school built up a database with all the music learners and their data. Help the music department with setting up the program to display the reports and queries that they require. | | |
|  |  |  |
| The **SelaMusicSchool.mdb** database contains one table called **MusicSchoolData**. | | |
|  |  |  |
| The **MusicSchoolData** table is structured with the following fields: | | |
|  |  |  |
|  | | |
|  |  |  |
| Example data from **MusicSchoolData** table: | | |
|  |  |  |
|  | | |
|  |  |  |
| **NOTE:**   * Connection code has been provided. * When the Restore Database button is clicked, the data in the database will be restored to the original data. * The name of the table to be used in your code must be **tblMusic**, which is a TADOTable object connected to the database. | | |
|  |  |  |
|  |  |  |
|  |  |  |
| Do the following:   * Compile and execute the program in the folder. The program currently has limited functionality. * Complete the code for each question as described. * Clear the **rich edit** each time before output is added. | | |
|  |  |  |
|  | | |
|  |  |  |
| 1.1 | Add code for **btnLearners** to display the total amount of learners on the rich edit component.    **Sample Output:** | (3) |
|  |  |  |
| 1.2 | Add code for the **btnGender** to display only the selected gender on the rich edit component. Display the name and surname of the learners, in neat columns.    **Sample Output:** |  |
|  | | |
|  |  |  |
|  | | |
|  |  | (9) |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 1.3 | Add code for the **btnRollOver** to do the year end procedure. This procedure requires the age of all the learners to be increased by one. Display a suitable message when the procedure is done with how many records were affected.    **Sample Output:** |  |
|  |  | (9) |
|  |  |  |
| 1.4 | Add code for the **btnInstrumentStat** that displays a count of the different instruments used by the learners.  **Sample Output:** |  |
|  |  | (12) |
|  |  |  |
| 1.5 | Add code for the **btnDeleteLearner** that uses an input box to obtain the **number** of the learner that needs to be deleted from the table. Display a suitable message if the learner was deleted.  **Sample Input:** |  |
|  |  |  |
|  |  |  |
|  | **Sample Output:** |  |
|  |  | (7) |
|  |  | **[40]** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **QUESTION 2: Create a Database** | | |
|  |  |  |
| 2.1 | The database name needs to be damsDB.mdb that will store information about the drivers and their exotic and rare cars. | (1) |
|  |  |  |
| 2.2 | The database needs to contains two tables, namely **tblDams** and **tblTowns**. |  |
|  |  |  |
| 2.2.1 | Create table **tblDams** with the following fields in it |  |
|  |  |  |
| |  |  |  | | --- | --- | --- | | FIELD | DATA TYPE | DESCRIPTION | | DamID (PK) | Number | Unique number assigned to the dam. | | DamName | Short Text | The name of the dam. | | River | Short Text | River that supplies the dam | | DamLevel | Number | The current level of water in the dam in thousands of liters. | | Capacity | Number | The total capacity of the dam in thousands of liters. | | HeightOfWall | Number | The height of the dam wall in metres. | | | |
|  |  | (6) |
|  |  |  |
| 2.2.2 | Create table **tbITowns** with the following fields in it |  |
|  |  |  |
| |  |  |  | | --- | --- | --- | | FIELD | DATA TYPE | DESCRIPTION | | TownID (PK) | Number | Unique number assigned to the town. | | TownName | Short Text | Name of the town. | | Province | Short Text | The province the town is located in. | | WaterRestrictions | Yes/No | Whether water restrictions have been imposed on the town. | | Population | Number | The population of the town. | | DamID | Number | The ID of the dam that supplies the town’s water. | | | |
|  |  | (6) |
|  |  |  |
| 2.2.3 | Create a one-to-many relationship with referential integrity in place between the two tables in the database: |  |
|  |  |  |
|  |  | (2) |
|  |  | **[15]** |

**TOTAL: 55 marks**