



Province of the  
**EASTERN CAPE**  
EDUCATION

**NATIONAL  
SENIOR CERTIFICATE**

**GRADE 12**

**SEPTEMBER 2019**

**INFORMATION TECHNOLOGY P1**

**MARKS: 150**

**TIME: 3 hours**

---

This question paper consists of 20 pages.

---

**INSTRUCTIONS AND INFORMATION**

1. This question paper is divided into FOUR sections. Candidates must answer ALL the questions in ALL FOUR sections.
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.
3. This question paper is set with programming terms that are specific to Delphi programming language.
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 locate, search, sort and selection, must be developed from first principles. You may NOT use the built-in features of Delphi 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 name appears as a comment in every program that you code, as well as on every event indicated.
11. If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.
12. At the end of this examination session you must hand in a disk/CD/DVD/flash disk with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you for this examination session. Make sure that all files can be read.

13. The files that you need to complete this question paper have been given to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **19JtMn&i**

Once extracted, the following list of files will be available in the folder **DataSept2019**:

**Question 1:**

Question1\_u.pas  
Question1\_u.dfm  
Question1\_p.dpr  
Question1\_p.res

**Question 2:**

ABC.jpg  
Bugs.jpg  
VideoClass.pas  
Question2\_u.pas  
Question2\_u.dfm  
Question2\_p.dpr  
Question2\_p.res

**Question 3:**

dbConnection\_u.pas  
VideoClub.mdb  
VideoClubBackUp.mdb  
Question3\_u.pas  
Question3\_u.dfm  
Question3\_p.dpr  
Question3\_p.res

**Question 4:**

Videos.txt  
Question4\_u.pas  
Question4\_u.dfm  
Question4\_p.dpr  
Question4\_p.res

## QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

- Open the incomplete program in the **Question 1** folder.
- Enter your full name as a comment in the first line of the **Question1\_u.pas** file.
- Compile and execute the program. The user interface displays two panels, **pnlQuestion1\_1** and **pnlQuestion1\_2**. The panel, **pnlQuestion1\_2**, is disabled before run time. The program has no functionality currently.
- Follow the instructions below to complete the code for each section of QUESTION 1, as described in QUESTION 1.1 and QUESTION 1.2.

### 1.1 Button [1.1 Join]

A learner at school needs to create a profile so that he/she can upload videos to be viewed on the school's intranet.

A learner must enter his/her e-mail address, password, grade and age. Each of these details must satisfy certain criteria before the learner can be registered to upload videos, i.e. the panel named **pnlQuestion1\_2** can then be enabled.

Write code in the **Q1\_1btnJoin** event handler to do the following:

- 1.1.1 Declare suitable variables and extract the input (name, e-mail, age, grade) from the input components.
- 1.1.2 Write code to test if the input satisfies the criteria as described below:
  - The e-mail address must contain the '@' symbol.
  - A valid password must satisfy the following criteria:
    - The password must consist of 8 or more characters.
    - 1 or more character must be a capital letter of the alphabet.
    - 2 or more characters must be a number from 0 to 9.
  - Only grade 10, grade 11 and grade 12 learners who are older than 15 years of age will be allowed to register and upload videos.
  - If all the above criteria are met, then write code to enable the panel named **pnlQuestion1\_2**.
  - If one of the above criteria is not met, then write code to disable the panel named **pnlQuestion1\_2**.

## 1.2 Button [1.2 Upload]

The size of a video is measured in Megabytes (Mb) and the speed of uploading a video is measured in Megabytes per second (Mb/s).

Learners will choose a category from the combobox, **cmbCategory**, and then they will enter the title and size of the video they wish to upload and the speed of their internet connection line in the edit boxes provided.

Write code to do the following:

- 1.2.1 Obtain the input from the input components.
- 1.2.2 The length of the video title must be a maximum of 23 characters. If the length is greater than 23 characters, then the edit box, **edtTitle**, must be cleared and the procedure must close.
- 1.2.3 Calculate how many minutes it will take to upload the video, using the details described below:
  - The time, in seconds, taken to upload a video is calculated by the formula:  
***Size of video / Speed of the line***
  - Convert the result of the formula above to the maximum number of seconds it will take to upload the video. (You must convert the seconds to the next whole number.)
    - Example: If a video is a size of 643.5 Mb and the speed of the line is 6 Mb/s, then the time taken to upload the video will be 107.25 seconds. This must be converted to 108 seconds.
    - Example: If a video is a size of 847 Mb and the speed of the line is 10 Mb/s, then the time taken to upload the video will be 84.7 seconds. This must be converted to 85 seconds.
    - Example: If a video is a size of 87 Mb and the speed of the line is 3 Mb/s, then the time taken to upload the video will be 29 seconds. This does not have to be converted because it is a whole number.
  - Calculate how many minutes it will take to upload the video.
- 1.2.4 A video code must be compiled using the following criteria:
  - A random number from 10 to 100 (both included)
  - The last character of the title of the video is added after the random number.
  - Lastly, the length of the title must be added to the video code.
  - Example of a video title: "A day in the park" / Video code = 67k17

- 1.2.5 All details of the video must be displayed in the richedit component, **redUploads**, in the following order:

*<Title of video> <Video Code> <Category> <Minutes to upload>*

- The minutes to upload must be formatted to 1 decimal place.
- Use the tab stops provided to allow for neat spacing.

**NOTE:** Headings for the richedit component have been provided in the code of the event handler.

Example of output:

Title of video	Code	Category	Minutes
At the park in March	62h20	Chess	27.3
June competition	88n16	Ballroom	14.3

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[37]

## QUESTION 2: OBJECT-ORIENTED PROGRAMMING

Your school's computer club has started to develop a program where learners can comment, like or dislike videos that are available for viewing. Each video must store the amount of likes and dislikes and all comments associated with that video. Each video must also be rated according to the number of likes and dislikes associated with it.

Do the following:

- Open the incomplete program in the Question 2 folder.
- Open the incomplete object class **VideoClass.pas**.
- Enter your name as a comment in both **VideoClass.pas** and **Question2\_u.pas**.
- Compile and execute the program. Currently the program has no functionality.
- Code has been written to load a picture file onto the image component provided and to display the panels provided. Do NOT remove or change any provided code.

The following user interface is displayed when the video, Bugs, has been chosen.

The screenshot shows a video player interface. At the top, there is a dropdown menu labeled 'Choose a video to watch' with 'ABC Bugs' selected. To the right of the dropdown is a button labeled '2.2.1 Watch this Video'. Below the dropdown, the video title 'Bugs' is displayed. To the right of the title is a 'RATING' section. Below the title, there are two buttons: '2.2.2 Like' with a thumbs up icon and '2.2.3 Dislike' with a thumbs down icon. The video player area shows a cat with the text: '99 little bugs in the code.', '99 little bugs in the code.', 'Take one down, patch it around.', and '127 little bugs in the code...'. Below the video player, there is a form with two input fields: 'Enter your name:' and 'Enter your comment below (Maximum of 30 characters ONLY)'. To the right of the form is a button labeled '2.2.4 Submit this Comment'. On the right side of the interface, there is a large empty box labeled 'Comments:'.

Complete the code for this program, as specified in QUESTION 2.1 and QUESTION 2.2.

- 2.1 The incomplete class (**TVideo**) contains the declaration of five attributes that describe the **objVideo** object.

NAMES OF ATTRIBUTES	DESCRIPTION
fvideoname	The name of the video
fcomments	A string containing a list of all comments including the name of the video, names of the learners who made the comments and the date of each comment
flikes	The number of likes for the video
fdislikes	The number of dislikes for the video
frating	The number of stars that the video is rated at

The following complete methods have been provided in the unit VideoClass.pas:

**addLike, addDislike, toString**

Complete the code in the object class, as described in QUESTION 2.1.1 to QUESTION 2.1.4 below.

- 2.1.1 Write code for a **constructor** method named **Create** that will receive the name of a video as a parameter.  
Assign this parameter value to the correct attribute and initialise the attributes for the number of likes and dislikes to zero. (4)
- 2.1.2 Write code for a method named **AddNewComment** that will receive two string parameters which will each represent a comment and a date.  
Set the value of the attribute **fcomments** as described below:
- The two parameters must be joined to the attribute **fcomments** :
- An empty line must be added before the new comment is joined to the attribute.
  - Add the new comment from the parameter list after the empty line is added.
  - Add a tab space and then add the date from the parameter list to the string.
  - An empty line must then be added after the date. (7)
- 2.1.3 Write a method named **SetRating** which will set a value to the attribute, **frating**, according to the following criteria:
- Subtract the number of dislikes from the number of likes.
  - If the difference is less than or equal to 0, then the rating must be 0.
  - The rating must be set to 5 for any value greater than 3, otherwise the rating must be set to 2. (7)
- 2.1.4 Write code for a method named **GetRating** which must return a string consisting of stars (\*).  
The string must contain stars (\*) joined together depending on the value of the attribute, **frating**. Example: If frating is 2 then the string will consist of 2 stars. (\*\*) (5)



2.2 An incomplete unit **Question2\_u.pas** has been provided.

It contains code for the object class to be accessible and has an object variable **objVideo** already declared. It also contains a variable to hold the system date.

**NOTE:** The system date is provided and saved in a string variable named **SystemDate** in the unit **Question2\_u.pas**

*Global variables supplied:* **objVideo: tvideo; SystemDate: string;**

Code to calculate the system date and to display the picture file chosen from the list box has been provided. Do NOT delete or change any provided code.

The user will choose a video to view and code is provided in the onclick event of the list box named **lbxVideos** to load a picture onto the image component and show the panels for adding likes, dislikes and comments.

Follow the instructions below to code the solution:

**2.2.1 OnClick event of the list box named lbxVideos (TQuestion2.lbxVideosClick)**

The user chooses a video name from the list box, **lbxVideos**.  
Write code to instantiate the object **objVideo** using the video name chosen.

(3)

**2.2.2 Button bitbtnLike [2.2.2 Like] and Button bitbtnDislike [2.2.2 Dislike]**

In the onclick event of **bitbtnLike**, use the method of the class named **addLike** to add a 'like' to the video object

In the onclick event of **bitbtnDislik**, use the method of the class named **addDislike** to add a 'dislike' to the video object.

(2)

**2.2.3 Button Q2\_2\_3btnView [2.2.3 Watch this video]**

Write code using a dialog box to allow the user to enter "Y" or "N" to the question: "Do you like this video?".

If the answer is "Y", then write code to execute the onclick event of the bitmap button named **bitbtnLike**, otherwise write code to execute the onclick event of the bitmap button named **bitbtnDislike**.

Add code using the methods of the class named **SetRating** and **GetRating** to update the rating and then display the updated rating in the panel named **pnlRatings**.

(6)

### 2.2.4 Button Q2\_2\_4btnSubmit [2.2.4 Submit this comment]

The user will enter a name and a comment in the edit boxes provided.

You must write code to join these two strings together with the comment on a separate line.

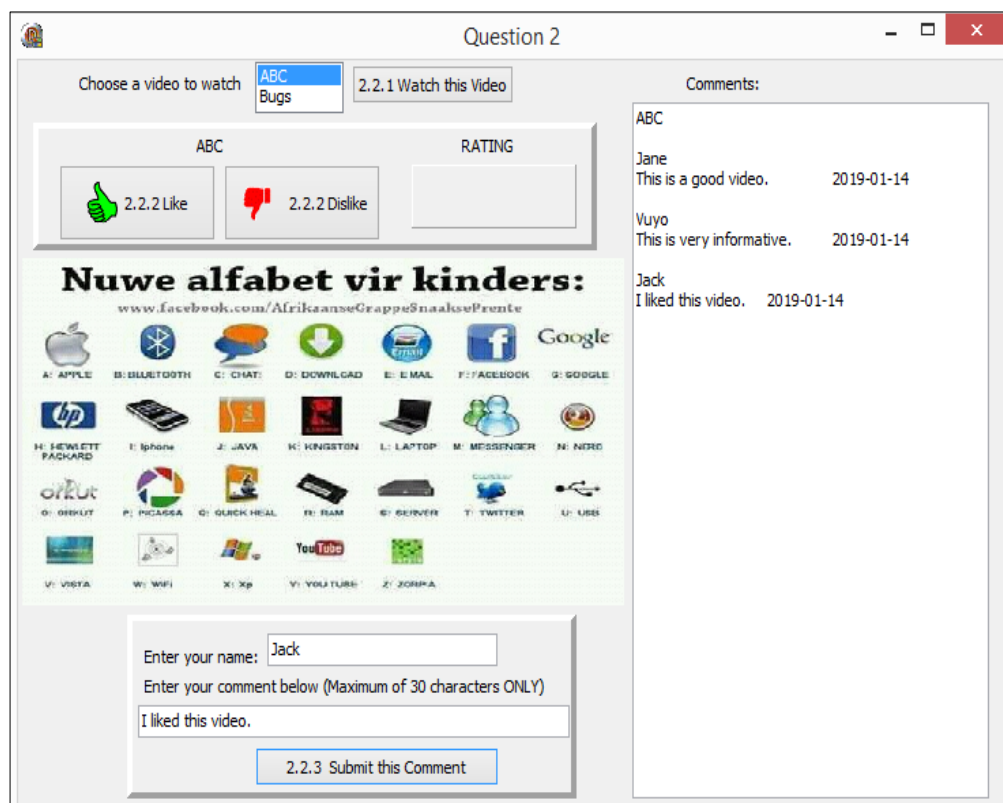
Example of compiled string: Jane Doe

This is a good video

Clear the richedit component named **redComments**.

Use the methods of the class named **addNewComment** (with the compiled string and system date as parameters) and **toString**, to update and display the comments in the richedit component named **redComments**.

Example of output for QUESTION 2.2.4:



(7)

- Enter your name and surname as a comment in the first line of the program file. (In both the class and the main program that uses the class)
- Save your programs.
- A printout of the code of both units may be required.

[41]

**QUESTION 3: DATABASE PROGRAMMING**

The database **VideoClub.mdb** contains the details of profile names of people who upload videos for viewing by members of a video club as well as all the videos belonging to each profile. The database contains two tables, namely **Profile** and **Video**.

Table: **Profile**

This table contains the profile name of a person who uploads videos and the date that their profile was created.

Field name	Data type	Description
ProfileID	Autonumber	A unique number assigned to each profile
ProfileName	Text (50)	The name of the profile
DateCreated	Date/Time	The date on which the profile was created

Example of data in the **Profile** table:

	ProfileID ▾	ProfileName ▾	DateCreated ▾
+	1	JanetandJohn	2016-12-15
+	2	TimmyTom	2015-09-30
+	3	MackyDee	2017-11-27
+	4	DonDoe	2015-07-27
+	5	SteveXolo	2015-10-27
+	6	SumiSam	2015-03-23

Table: **Video**

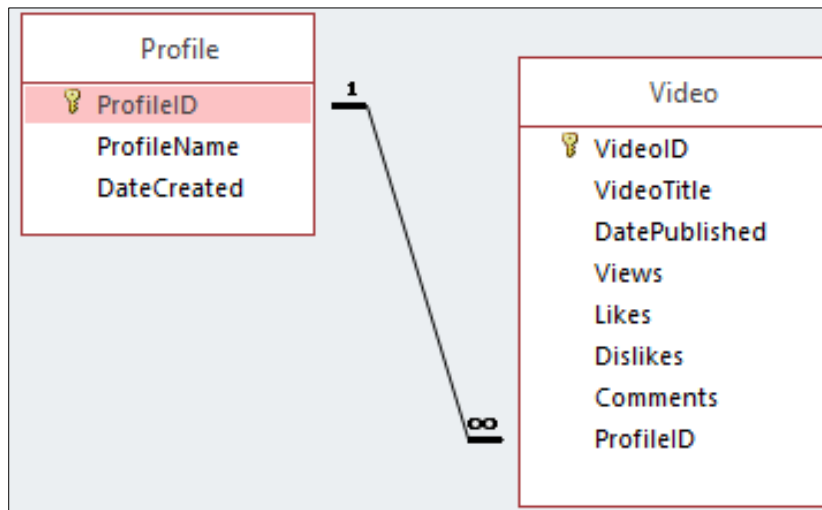
This table contains all the videos that have been uploaded by the profiles.

Field name	Data type	Description
VideoID	Text (50)	A unique code assigned to the video
VideoTitle	Text (80)	The title of the video
DatePublished	Date/Time	The date that the video was published
Views	Number	The number of views for the video
Likes	Number	The number of likes for the video
Dislikes	Number	The number of dislikes for the video
Comments	Number	The number of comments for the video
ProfileID	Number	The number that identifies the profile that the video belongs to

Example of data of the first ten records of the **Video** table.

VideoID	VideoTitle	DatePublished	Views	Likes	Dislikes	Comments	ProfileID
svfFvkHgeWY	#TimsDark Experiment   Tim Hortons Dark Roast Coffee	2014-08-15	2706	1479	136	275	2
ag-Mu3jHfWU	#TimsNextDoor – Serving the Neighbourhood	2014-09-26	2356	509	27	70	2
tPFomA-Gm84	\$1 Cravings Menu Haul   Taco Bell	2014-08-15	770	144	83	106	1
fLa4miaiXIA	Belluminati - \$1 Stacker   Commercial   Taco Bell	2017-12-15	4023	316	319	254	1
KelYyRvV0go	Brain Freeze - Episode 5   Taco Tales Season 1   Taco Bell	2017-01-05	4095	394	11	98	1
IJKUcn7Cy5A	Celebrating 50 Years of SUBWAY Restaurants®	2015-08-26	656	480	160	0	6
oludwaDrmfg	Chrissy Teigen, Alexa Losey, and Andrew W.K	2014-01-14	1153	931	239	205	1
9XzNogAjlFw	Congrats Sid! 2016 NHL Stanley Cup Winner	2016-06-13	635	411	46	61	2
S6xjRrY3CWA	Dan's Coffee Run	2013-02-08	2778	326	4	32	5
HTOpqgQx0xc	Dan's Everylove Story	2012-02-06	280	134	15	118	5

The following one-to-many relationship with referential integrity exists between the two tables in the database:



Do the following:

- Open the incomplete project file called **Question3\_p.dpr** in the **Question 3** folder.
- Enter your name as a comment in the first line of the **Question3\_u.pas** unit file.
- Compile and execute the program. The program has no functionality currently.

The user interface is displayed on the next page.

- Follow instructions to complete the code for each question, as described in QUESTION 3.1 and QUESTION 3.2.
- **Use SQL code to answer QUESTION 3.1 and Delphi code to answer QUESTION 3.2.**

Question 3.1 Display area (SQL RESULTS)

3.1.1 3.1.2 3.1.3 3.1.4 3.1.5

VIDEOS

VideoID	VideoTitle	DatePublished	Views	Likes	Dislikes	Comments	ProfileID
svfFvkHgeWY	#TimsDark Experiment   Tim Hortons Dark Roast Coffee	2014-08-15	2706	1479	136	275	2
ag-Mu3jHfWU	#TimsNextDoor – Serving the Neighbourhood	2014-09-26	2356	509	27	70	2
tPFomA-Gm84	\$1 Cravings Menu Haul   Taco Bell	2014-08-15	770	144	83	106	1
fla4miaiXIA	Belluminati - \$1 Stacker   Commercial   Taco Bell	2017-12-15	4023	316	319	254	1
KeYyRvV0go	Brain Freeze - Episode 5   Taco Tales Season 1   Taco Bell	2017-01-05	4095	394	11	98	1
IJKUcn7Cy5A	Celebrating 50 Years of SUBWAY Restaurants®	2015-08-26	656	480	160	0	6

PROFILES

ProfileID	ProfileName	DateCreated
1	JanetandJohn	2016-12-15
2	TimmyTom	2015-09-30
3	MackyDee	2017-11-27
4	DonDoe	2015-07-27
5	SteveXolo	2015-10-27
6	SumiSam	2015-03-23

Question 3.2 Display area

3.2.1 3.2.2

Restore Database

**NOTE:**

- The **[Restore Database]** button is provided to restore your data contained in the database to the original content. If you need to test your code on the original data, you may click this button to restore data.
- Do NOT change any of the code provided.
- Code is provided to link the GUI components to the database.
- TWO variables are declared as global variables, as described in the table below.

Variable	Data type	Description
tblProfiles	TADOTable	Refers to the table named <b>Profile</b>
tblVideos	TADOTable	Refers to the table named <b>Video</b>

**3.1 In this section you may ONLY use SQL statements to answer QUESTION 3.1.1 to QUESTION 3.1.5.**

Code to execute the SQL statements and display the results of the queries is provided. The SQL statements are incomplete.

The section of the user interface for SQL questions is displayed below:

Question 3.1 Display area (SQL RESULTS)

3.1.1 3.1.2 3.1.3 3.1.4 3.1.5

Do the following to complete the incomplete SQL statements assigned to the variables sSQL1, sSQL2, sSQL3, sSQL4 and sSQL5 per question respectively.

### 3.1.1 Button [3.1.1]

Write SQL code to display the **DateCreated** and the **ProfileName** of all profiles sorted from the newest to the oldest profile created.

Example of output:

DateCreated	ProfileName
2017-11-27	MackyDee
2016-12-15	JanetandJohn
2015-10-27	SteveXolo
2015-09-30	TimmyTom
2015-07-27	DonDoe
2015-03-23	SumiSam

(4)

### 3.1.2 Button [3.1.2]

The user must enter the name of a profile. Code has been provided for the profile name “**DonDoe**” to be entered in an input box and saved in a variable named **sline**.

Write SQL code to display the **DatePublished** and **earnings** of all videos that have been uploaded by the profile name stored in the variable **sline**. Both tables must be used.

The **earnings** are calculated by subtracting the number of dislikes from the number of likes and adding the number of comments. The result of this calculation is then divided by the number of views.

**Earnings = (Likes – Dislikes + Comments) / Views.**

The earnings must be formatted as currency to two decimal places and displayed as the heading **Earnings**.

Example of output:

DatePublished	Earnings
2015-06-08	R 0.34
2017-08-21	R 0.50
2016-06-01	R 0.31
2015-07-21	R 0.27
2015-07-07	R 0.25

(9)

### 3.1.3 Button [3.1.3]

Write an SQL statement to display video titles [**VideoTitle**] where the number of views is greater than or equal to 1000 and the number of dislikes is less than or equal to 10.

Example of output of the first five records:

VideoTitle
Tim Hortons Drive Thru Rookies with Sid & Nate C
Starbucks Joymakers
Mary J. Blige and David Foster Perform at Seattle Store
Tim Hortons Drive Thru Rookies with Sid & Nate   Surprise
Tim Hortons Drive Thru Rookies with Sid & Nate E

(4)

### 3.1.4 Button [3.1.4]

Write an SQL statement to delete the video that was published in the year 2008.

(Code has been written to display the successfully updated table after deletion, in order of DatePublished from oldest to newest.)

Example of output:

DatePublished	VideoTitle
2010-05-04	Your Perfect Starbucks Latte
2011-04-17	One Person Can Save Trees, Together We Can Save Forests
2011-12-17	Happy Holidays from Starbucks!
2012-02-06	Dan's Everylove Story
2012-02-06	Starbucks Cup Magic for Valentine's Day

(4)

### 3.1.5 Button [3.1.5]

Write an SQL statement to change the name of the profile named **TimmyTom** to **Tim Horton**.

(Code has been written to display the successful results.)

Example of output:

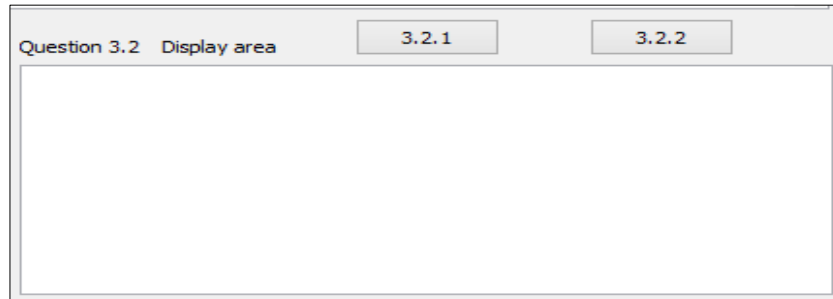
ProfileID	ProfileName	DateCreated
1	JanetandJohn	2016-12-15
2	Tim Horton	2015-09-30
3	MackyDee	2017-11-27
4	DonDoe	2015-07-27
5	SteveXolo	2015-10-27
6	SumiSam	2015-03-23

(3)

**3.2 In this section, only Delphi programming code may be used to answer QUESTION 3.2.1 and QUESTION 3.2.2.**

NO marks will be awarded for SQL statements in QUESTION 3.2.

The section of the user interface for QUESTION 3.2 is shown below:



**3.2.1 Button [3.2.1]**

Write code to display all video titles in the richedit, **redDisplay**, that contain the single word **'the'** in the video title.

Example of output:

```
#TimsNextDoor – Serving the Neighbourhood
Future Father Of The Year - Episode 4 | Taco Tales Season 1
Jump the Boards with Sidney Crosby and Tim Hortons
Looking to watch the Solar Eclipse? ✱
McDonald's: Behold, The Micro Mac
National Taco Day: Glen And The Magic Taco | Taco Bell
The Belluminati | Commercial | Taco Bell
The Great Escape - Episode 3 | Taco Tales Season 1
The Starbucks Reserve Roastery and Tasting Room
The World's Fastest Dunkin' Run Wingsuit Commercial :60
Tim Hortons #WarmWishes | The Cup of Good Deeds
```

(6)

**3.2.2 Button [3.2.2]**

Write code to delete all videos that have more than 8 dislikes.

Example of output of all records displayed in the database grid, **dbgVideos**:

VIDEOS								
VideoID	VideoTitle	DatePublished	Views	Likes	Dislikes	Comments	ProfileID	
▶ S6xjRrY3CWA	Dan's Coffee Run	2013-02-08	2778	326	4	32	5	
R5YdGpUsRwg	Hippo Campus breezes through a spirited version of "South"	2015-08-24	1217	374	0	10	4	
8nvqQzjq10w	Starbucks Cup Magic for Valentine's Day	2012-02-06	2112	366	8	19	5	
qQvZsnl-im8	Tim Hortons Drive Thru Rookies with Sid & Nate   Surprise	2015-09-24	2405	386	8	14	2	
AfloXibCCG0	Tim Hortons Drive Thru Rookies with Sid & Nate E	2015-09-24	2440	374	7	18	2	
8i_o6dKebpk	Tims Run with Sid & Nate   Game On	2016-09-07	589	334	7	26	2	

(6)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[36]



**QUESTION 4: PROBLEM-SOLVING PROGRAMMING**

Do the following:

- Open the incomplete program in the Question 4 folder.
- Enter your name and surname as a comment in the first line of the Question4\_u.pas file.
- Compile and execute the program. Currently the program has no functionality.

Supplied GUI:

The supplied GUI represents the interface of the program used by the school's computer club to analyse and make changes to videos and profiles.

A text file is included which contains the profile names, video codes and the length of each video in seconds.

The format of the text file is:

<Profile name><Video code><length of video in seconds>

Code has been written to read only two of the three items from each line of the text file into 2 global, parallel arrays, named **arrProfiles** and **arrMinutes**.

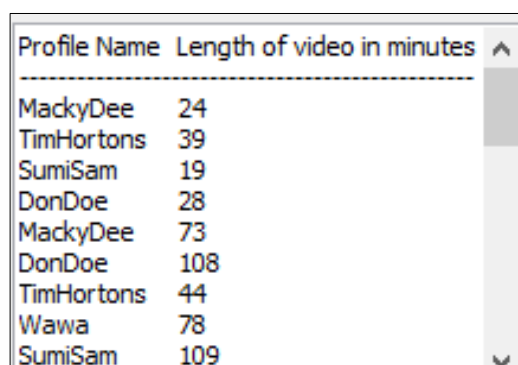
**ArrMinutes** contains the duration of each video as integers. These were calculated from the seconds converted to minutes and then rounded to integers.

**ArrProfiles** contains strings representing the profile names

The variable **icountarr** holds the total number of items for the two parallel arrays.

Code has been written in the **OnActivate** event of the form to read data from the text file into the two arrays and then display the array contents into the rich edit.

When the program is run, the two arrays are displayed in the richedit as shown in the diagram below:



Profile Name	Length of video in minutes
MackyDee	24
TimHortons	39
SumiSam	19
DonDoe	28
MackyDee	73
DonDoe	108
TimHortons	44
Wawa	78
SumiSam	109

Complete the code for each question, QUESTION 4.1 and QUESTION 4.2.

**NOTE:**

- Good programming techniques and modular design must be applied in the design and coding of your solution.
- You may NOT change the code provided.

#### 4.1 Menu option [4.1 Sort and Remove]

- 4.1.1 Sort both parallel arrays, **arrProfiles** and **arrMinutes**, in alphabetical order of the profile names.

Below is a display of the result of sorting the arrays in alphabetical order of profile name:

Output area for Question 4.1	
Profile Name	Length of video in minutes
DonDoe	28
DonDoe	108
DonDoe	33
DonDoe	68
DonDoe	245
DonDoe	54
DonDoe	51
MackyDee	42
MackyDee	73
MackyDee	24
MackyDee	124
SteveXolo	20
SteveXolo	47

- 4.1.2 Remove all details of the profile name “**DonDoe**” from both arrays, **arrProfiles** and **arrMinutes**.
- 4.1.3 Create a text file using the file name **Revised.txt** and write all contents of the two sorted and revised arrays to the text file in the following order:

*< Profile name><semicolon><length of video in minutes>*

4.1 Example of the output to a text file:

```

Revised.txt - Notepad
File Edit Format View Help
MackyDee;42
MackyDee;73
MackyDee;24
MackyDee;124
SteveXolo;20
SteveXolo;47
SteveXolo;32
SteveXolo;52
SteveXolo;32
SumiSam;109
SumiSam;75
SumiSam;76
SumiSam;148
SumiSam;19
SumiSam;39

```

(17)

## 4.2 Menu option [4.2 Advertisements]

The school's computer club has obtained funding from advertising companies. Each video belonging to learners who have profiles may have 2 advertisements or 4 advertisements placed in positions during all of their videos' playback times.

The diagram below shows the stringgrid on the form when the program is run.

Videos	1st Position	2nd Position	3rd Position	Last Position
1	No advert	No advert	No advert	No advert
2	No advert	No advert	No advert	No advert
3	No advert	No advert	No advert	No advert
4	No advert	No advert	No advert	No advert
5	No advert	No advert	No advert	No advert
6	No advert	No advert	No advert	No advert
7	No advert	No advert	No advert	No advert
8	No advert	No advert	No advert	No advert

The contents of the global 2-dimensional array are displayed in the stringgrid component in the onactivate event of the form using the **Display** procedure which has been written for you.

A global 2-dimensional array named **arr2D** containing real data types has been supplied.

In this question you must fill the two-dimensional array named **arr2D** with data representing the positions that adverts will occur during the playback of videos. You must then call the **Display** procedure to display the contents of the array **arr2D** in the stringgrid.

No advertisement will occur at the start of the playback of the video.

The first advert will take place at  $\frac{1}{4}$  of the way during playback.

The second advert will take place at half way through the video during playback.

The third advert will take place at the  $\frac{3}{4}$  point of the video during playback.

The last (fourth) advert will take place when the video stops playing.

The user will choose a profile name from the combobox provided.

Code has been provided for the user to enter the number of advertisements required into an input box component for the selected profile name. This quantity is stored in a local variable name **inumber**.

Only 2 advertisements or 4 advertisements can be chosen for a profile name.

The profile name chosen will then have that amount of advertisements running through all his/her videos.

The two-dimensional array must then be filled with position times of advertisement placements of all videos belonging to that chosen profile name.

If **two** advertisements are chosen for a profile, then the advertisements must appear at the first quarter point of the video during playback and then at the third quarter point of the video during playback.

If **four** advertisements are chosen for a profile, then the advertisements must appear at the first quarter point of the video during playback and then at the half way point of the video playback and again at the third quarter point of the video during playback and lastly, at the end of the video during playback.

- All contents of the array named **arr2D** must first be initialised to 0 before new data can be stored in the array.
- The advertisement placement times for each video must then be calculated and stored in the array named **arr2D**.
- The array, **arr2D**, must then be displayed in the stringgrid using the procedure **Display** provided.

*4.2 Example of output if SteveXolo is chosen and 4 advertisements are required:*

Videos	1st Position	2nd Position	3rd Position	Last Position
1	5.0	10.0	15.0	20.0
2	11.8	23.5	35.3	47.0
3	8.0	16.0	24.0	32.0
4	13.0	26.0	39.0	52.0
5	8.0	16.0	24.0	32.0
6	No advert	No advert	No advert	No advert
7	No advert	No advert	No advert	No advert
8	No advert	No advert	No advert	No advert

*4.2 Example of output if SteveXolo is chosen and 2 advertisements are required:*

Videos	1st Position	2nd Position	3rd Position	Last Position
1	5.0	No advert	15.0	No advert
2	11.8	No advert	35.3	No advert
3	8.0	No advert	24.0	No advert
4	13.0	No advert	39.0	No advert
5	8.0	No advert	24.0	No advert
6	No advert	No advert	No advert	No advert
7	No advert	No advert	No advert	No advert
8	No advert	No advert	No advert	No advert

(19)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

[36]

**TOTAL: 150**