HOËRSKOOL BRANDWAG

November 2022

Eksamen

Graad 11

Inligtingstegnologie V1

Punte: 150
Tyd: 3 uur

Eksaminator: A van Rooyen
Moderator: W du Plessis

Hierdie vraestel bestaan uit 19 bladsye.

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. Answer ALL the questions.

3. You require the files listed below to answer the questions. They are on the network folder linked to your examination username:

QUESTION1:	**QUESTION2:**	**QUESTION3:**
frmQuestion1_P.dpr	frmQuestion2_P.dpr	frmQuestion3_P.dpr
frmQuestion1_U.dfm	frmQuestion2_U.dfm	frmQuestion3_U.dfm
frmQuestion1_U.pas	frmQuestion2_U.pas	frmQuestion3_U.pas
frmQuestion1_U.res	frmQuestion2_U.res	frmQuestion3_U.res
frmQuestion1_p_Icon.ico	ConnectDB_U.pas	frmQuestion3_p_Icon.ico
	DefinitionsDB.mdb	Decimal.txt
	DB_Backup.mdb	
	frmQuestion2_P_Icon.ico	

4. Save your work at regular intervals as a precaution against power failures.

5. Rename the folder **11ITNovemberSurnameName** with your own. For example: 11ITNovember**SmithJohn**

6. Type your name and surname as a comment in the first line of each program.

7. Make sure you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.

8. 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 the data validation.

9. 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.

10. Routines such as 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.

11. All data structures must be defined by you, the programmer, unless the data structures are supplied.

12. After the examination you will be given time to print all your units to hand in.

INSTRUKSIES EN INLIGTING

1. Dit is 'n **drie-uur** eksamen. Weens die aard van hierdie eksamen is dit belangrik om daarop te let dat jy nie toegelaat sal word om die eksamenlokaal voor die einde van die eksamensessie te verlaat nie.

2. Beantwoord AL die vrae.

3. Jy benodig die lêers wat hieronder gelys word om die vrae te beantwoord. Hulle is op die netwerklêergids wat aan jou eksamengebruikersnaam gekoppel is:

VRAAG 1:	**VRAAG 2:**	**VRAAG 3:**
frmQuestion1_P.dpr	frmQuestion2_P.dpr	frmQuestion3_P.dpr
frmQuestion1_U.dfm	frmQuestion2_U.dfm	frmQuestion3_U.dfm
frmQuestion1_U.pas	frmQuestion2_U.pas	frmQuestion3_U.pas
frmQuestion1_U.res	frmQuestion2_U.res	frmQuestion3_U.res
frmQuestion1_p_Icon.ico	ConnectDB_U.pas	frmQuestion3_p_Icon.ico
	DefinitionsDB.mdb	Decimal.txt
	DB_Backup.mdb	
	frmQuestion2_P_Icon.ico	

4. Stoor jou werk met gereelde tussenposes as 'n voorsorgmaatreël teen kragonderbrekings.

5. Hernoem die lêergids **11ITNovemberVannaam** met jou eie. Byvoorbeeld: 11ITNovember**SmithJohn**

6. Tik jou naam en van as kommentaar in die eerste reël van elke program.

7. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal toegeken word volgens die gestelde vereistes.

8. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, as die vraag nie vir datavalidering vra nie, sal geen punte vir die datavalidering toegeken word nie.

9. Jou programme moet so gekodeer word dat hulle met enige data sal werk en nie net die voorbeelddata wat verskaf word of enige data-uittreksels wat in die vraestel verskyn nie.

10. Roetines soos soek, sorteer en seleksie moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude kenmerke van Delphi vir enige van hierdie roetines gebruik nie.

11. Alle datastrukture moet deur u, die programmeerder, gedefinieer word, tensy die datastrukture verskaf word.

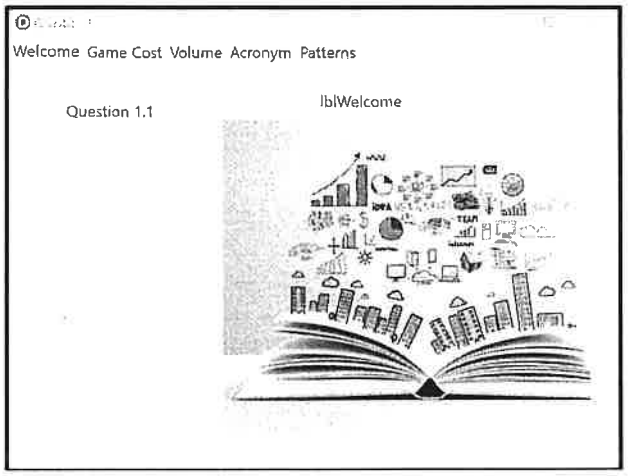
12. Aan die einde van die eksamen sal jy tyd kry om die eenhede te druk vir inhandiging.

QUESTION 1: GENERAL PROGRAMMING SKILLS

Do the following:

- Open the incomplete program in the folder **Question1**.
- Enter your name and surname as a comment in the first line of the **frmQuestion1_U.pas** file.
- Compile and execute the program. The user interface displays FIVE tab sheets.
- Follow the instructions below to complete the code for EACH section of QUESTION 1, as described in QUESTION 1.1. to QUESTION 1.5.

Example of the graphical user interface (GUI):



1.1. **Tab sheet [Welcome]:**

Button [Question 1.1]

Write code to do the following:

- Change the Image **imgPic** as follows:
  - Set the stretch property to true.
  - Set the width of the image to 377.
- Change the Label **lblWelcome** as follows:
  - Display the heading 'Life Long Learning'.
  - Underline the heading.
  - Change the font to 'Elephant'.

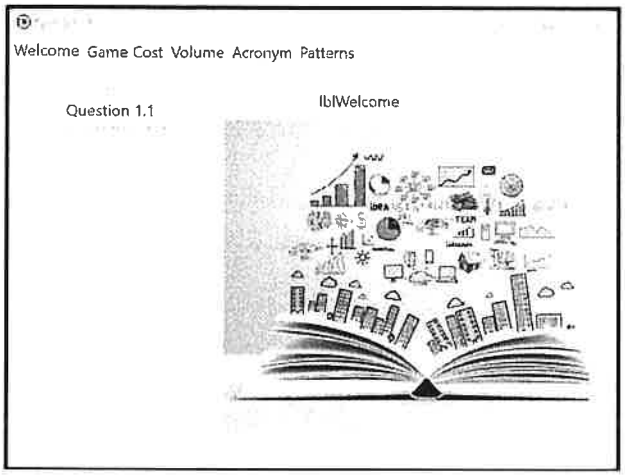
Example of output: (8)

VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE

Doen die volgende:

- Maak die onvolledige program oop in die lêergids **Question1**.
- Tik jou naam en van as kommentaar in die eerste reël van die **frmQuestion1_U.pas**-lêer.
- Stel die program saam en voer dit uit. Die gebruikerskoppelvlak vertoon VYF oortjieblaaie("tab sheets").
- Volg die instruksies hieronder om die kode vir ELKE afdeling van VRAAG 1 te voltooi, soos beskryf in VRAAG 1.1. tot VRAAG 1.5.

Voorbeeld van die grafiese gebruikerskoppelvlak (GUI):



1.1. **Oortjieblad [Welcome]:**

Knoppie [Question 1.1]

Skryf kode om die volgende te doen:

- Verander die Image **imgPic** soos volg:
  - Stel die stretch-eienskap op waar.
  - Stel die breedte van die prent op 377.
- Verander die etiket **lblWelcome** soos volg:
  - Vertoon die opskrif 'Life Long Learning'.
  - Onderstreep die opskrif.
  - Verander die lettertipe na 'Elephant'.

Voorbeeld van afvoer: (8)

1.2. **Tab sheet [Game Cost]:**

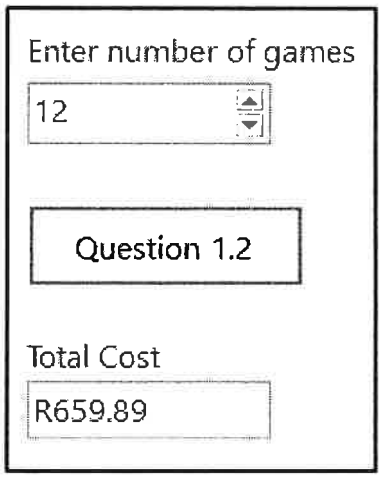
Button [Question 1.2]

Educational games are expensive but when users buy games in bulk they can qualify for a discount.

Write code to do the following:

- Store input from the user for the number of games they would like to purchase in the provided variable **iGames**.
- Determine the cost of the games as follows and store it in the variable provided **rCost**:
  - For every 10 games purchased the user gets one game for free.
  - Use the constant variable provided, holding the cost per game, to determine the final cost.
- Display the final cost in **edtCost**.

Example of output when **12** games was entered:



(5)

Oortjieblad [Game Cost]:

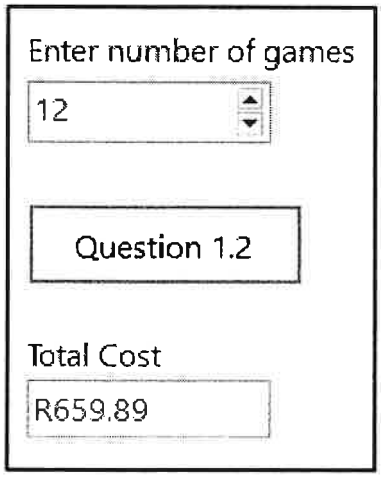
Knoppie [Question 1.2]

Opvoedkundige speletjies is duur, maar wanneer gebruikers speletjies in grootmaat koop, kan hulle vir 'n afslag kwalifiseer.

Skryf kode om die volgende te doen:

- Stoor toevoere van die gebruiker vir die aantal speletjies wat hulle wil koop in die gegewe veranderlike **iGames**.
- Bepaal die koste van die speletjies soos volg en stoor dit in die veranderlike **rCost**:
  - Vir elke 10 speletjies wat gekoop word, kry die gebruiker een speletjie gratis.
  - Gebruik die konstante veranderlike wat gegee word, met die koste per speletjie, om die finale koste te bepaal.
- Vertoon die finale koste in **edtCost**.

Voorbeeld van afvoer wanneer **12** speletjies aangevra is:

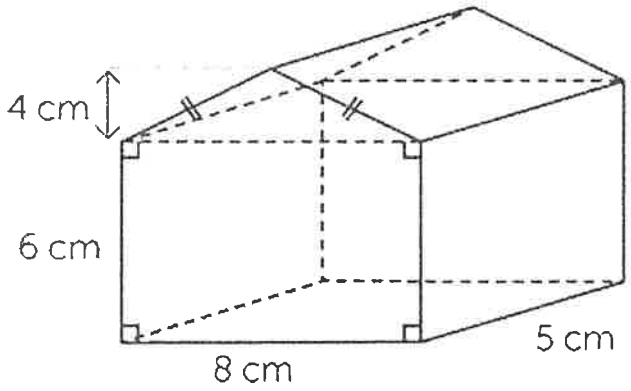


(5)

1.3. **Tab sheet [Volume]:**

Button [Question 1.3]

A mathematical game needs to determine the volume of a model house consisting of a triangular prism as the roof and a rectangular prism as the base of the house.



Declarations of the variables needed have been provided. Code has been provided to assign a value of 6 to the height of the house.

Write code to do the following:

- Make use of dialog boxes to store the input from the user for the length and width of the base of the model house and the height of just the roof.
- Determine the volume of the roof and the entire house making use of the following formulas:
  - Volume of triangular prism:
    1 / 2 X Roof Height X Width X Length
  - Volume of rectangular prism:
    Length X Width X House Height
- Calculate and display the percentage that the volume of the roof is of the total house volume.
- Display the volume of the house and the roof percentage, both formatted to ONE decimal place in the rich edit **redOutput**.

Example of output when **3** was entered as the height of the roof and **7** and **9** were entered for the width and length of the house respectively.

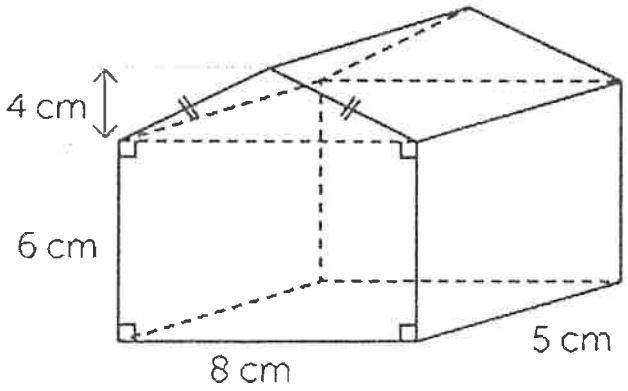
```
Total Volume = 472.5
Percentage = 20.0%
```

(10)

1.2. **Tabblad [Volume]:**

Knoppie [Question 1.3]

'n Wiskundige speletjie moet die volume van 'n modelhuis bepaal wat bestaan uit 'n driehoekige prisma as die dak en 'n reghoekige prisma as die basis van die huis.



Verklarings van die veranderlikes wat benodig word, is verskaf. Kode is verskaf om 'n waarde van 6 aan die hoogte van die huis toe te ken.

Skryf kode om die volgende te doen:

- Maak gebruik van dialoogkassies("input boxes") om die toevoere van die gebruiker vir die lengte en breedte van die basis van die modelhuis en die hoogte van net die dak te stoor.
- Bepaal die volume van die dak **en** die hele huis deur die volgende formules te gebruik:
  - Volume van driehoekige prisma:
    1 / 2 X Dak Hoogte X Breedte X Lengte
  - Volume van reghoekige prisma:
    Lengte X Breedte X Huis Hoogte
- Bereken en vertoon die persentasie wat die volume van die dak van die totale huisvolume is.
- Vertoon die volume van die huis en die dakpersentasie, beide geformateer tot EEN desimale plek in die rich edit **redOutput**.

  Voorbeeld van afvoer wanneer **3** as die hoogte van die dak ingevoer is en **7** en **9** vir die breedte en lengte van die huis onderskeidelik ingevoer is.

Total Volume = 472.5 Percentage = 20.0%

(10)

1.4. **Tab sheet [Acronym]:**

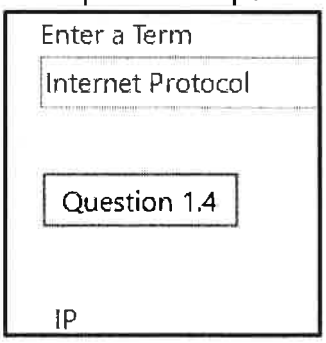
Button [Question 1.4]

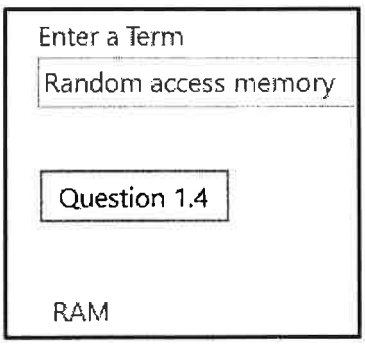
A program is needed to create acronyms for IT theory terminology. An *acronym* is an abbreviation formed by using the first letter of every word in a term.

Write code to do the following:

- Store input from the user for the term entered in **edtTerm**.
- Make use of a loop to create an acronym by extracting the first letter of every word in the term. Ensure that the acronym created is capitalised.
- Display the acronym in the label provided.

Examples of output:





(9)

1.3. **Tabblad [Acronym]:**

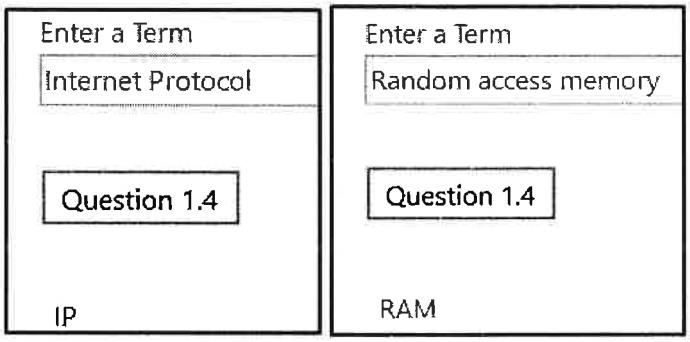
Knoppie [Question 1.4]

'n Program is nodig om akronieme vir IT-teorie-terminologie te skep. 'n Akroniem is 'n afkorting wat gevorm word deur die eerste letter van elke woord in 'n term te gebruik.

Skryf kode om die volgende te doen:

- Stoor die toevoer van die gebruiker vir die term wat in **edtTerm** ingelees is.
- Maak gebruik van 'n lus om 'n akroniem te skep deur die eerste letter van elke woord in die term te onttrek. Maak seker dat die akroniem wat geskep is, in hoofletters is.
- Vertoon die akroniem in die label wat voorsien word.

Voorbeelde van afvoer:



(9)

1.5. **Tab sheet [Collatz]:**

Button [Question 1.5]

The Collatz conjecture states that if you take any positive integer and repeatedly apply the following steps to find the next number;

- if the number is even, divide it by 2,
- if the number is odd, multiply it by 3 and add 1

the sequence of numbers will always reach 1.

A number for which this is not true has never been found, but the conjecture is has never been mathematically proven to be true.

A program would make it easier to test the conjecture on multiple values.

Write code to do the following:

- The user must input a number. Validate that this input is an integer number. If the input is invalid, write code to:
  - Clear the edit.
  - Place the cursor in the edit.
  - Change the colour of the edit to yellow.
  - Do not allow the rest of the code to execute.

- Continue to apply the calculations described below to determine the sequence until the number equals one. The number referred to below would first be the original number entered and thereafter it would be the answer to the previous calculation:
  - If the number is even, divide it by 2 to determine the next number.
  - If the number is odd then multiply it by 3 and add 1 to determine the next number.

- Count how many numbers are generated before a value of one is reached.
- Display each new term and the count in the rich edit **redCollatz**.

Example of output when **6** was entered:

```
1: 6
2: 3
3: 10
4: 5
5: 16
6: 8
7: 4
8: 2
9: 1
```

An **extract** of output when **259** was entered:

```
113: 160
114: 80
115: 40
116: 20
117: 10
118: 5
119: 16
120: 8
121: 4
122: 2
123: 1
```

(18)

1.4. **Tabblad [Collatz]:**

Knoppie [Question 1.5]

Die Collatz-veronderstelling bepaal dat as jy enige positiewe heelgetal neem en herhaaldelik die volgende stappe toepas om die volgende getal te vind;

- as die getal ewe is, deel dit deur 2,
- as die getal onewe is, vermenigvuldig dit met 3 en tel 1 by

dan sal die volgorde van getalle altyd 1 bereik.

'n Getal waarvoor dit nie waar is nie, is nog nooit gevind nie, maar die vermoede is nog nooit wiskundig as waar bewys nie.

'n Program sal dit makliker maak om die vermoede op veelvuldige waardes te toets.

Skryf kode om die volgende te doen:

- Die gebruiker moet 'n nommer inlees. Bevestig dat hierdie getal 'n heelgetal is. As die toevoer ongeldig is, skryf kode om die volgende te doen:
  - Maak die edit skoon.
  - Plaas die muiswyser in die edit.
  - Verander die kleur van die edit na geel.
  - Moenie toelaat dat die res van die kode uitgevoer word nie.

- Gaan voort om die berekeninge wat hieronder beskryf word, toe te pas om die reeks te bepaal totdat die getal gelyk is aan 1. Die nommer waarna hieronder verwys word, sal eers die oorspronklike nommer wees wat ingelees is en daarna sal dit die antwoord op die vorige berekening wees:
  - As die getal ewe is, deel dit deur 2 om die volgende getal te bepaal.
  - As die getal onewe is, vermenigvuldig dit dan met 3 en tel 1 by om die volgende getal te bepaal.

- Tel hoeveel getalle gegenereer word voordat 'n waarde van een bereik word.
- Vertoon elke nuwe term en die telling in die rich edit **redCollatz**.

Voorbeeld van afvoer wanneer **6** ingelees is:

'n Uittreksel van afvoer wanneer **259** ingelees is:

- Enter your Name and Surname as a comment in the first line of the program file **frmQuestion1_U**.
- Save all the files ('File/Save All').
- Printouts of the code for the unit (**frmQuestion1_U**) will be required.

TOTAL QUESTION 1: 50

1: 6 2: 3 3: 10 4: 5 5: 16 6: 8 7: 4 8: 2 9: 1

113: 160 114: 80 115: 40 116: 20 117: 10 118: 5 119: 16 120: 8 121: 4 122: 2 123: 1

(18)

- Tik jou Naam en Van as kommentaar in die eerste reël van die programlêer **frmVraag1_U**.
- Stoor al die lêers ('File/Save All').

TOTAAL VRAAG 1: 50

QUESTION 2: DATABASE PROGRAMMING

SCENARIO: The database **DefinitionsDB** contains a collection of IT theory terms and their definitions. The database contains two tables namely **tblTopics** and **tblTerms**. A program is needed to enable a user to view and edit the contents of the database.

The data pages attached at the end of the question paper provide information on the design of the database and its contents.

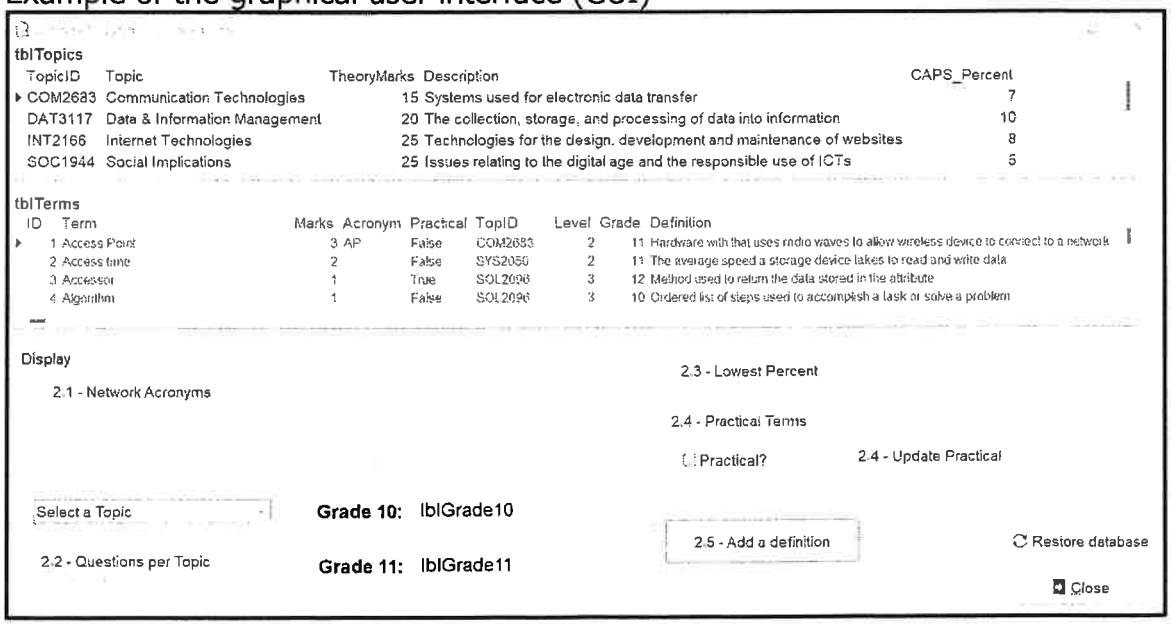
Do the following:

- Open the incomplete project file **frmQuestion2_P.dpr** in the folder **Question2**.
- Enter your name and surname as a comment on the first line of the **frmQuestion2_U.pas** file.
- The **Restore database** button is provided to restore the data contained in the database to the original content.
- Follow the instructions to complete the code for each question as described below.
- Do not change any of the provided code.
- The database has been linked to the GUI components for you.
- TWO variables are declared as public variables, as described in the table below.

Variable	Data type	Description
tblTopics	TADOTable	Refers to the database table tblTopics
tblTerms	TADOTable	Refers to the database table tblTerms

2. In this question only Delphi programming code may be used to answer QUESTION 2.1 to QUESTION 2.5

Example of the graphical user interface (GUI)



VRAAG 2: DATABASISPROGRAMMERING

SCENARIO: Die databasis **DefinitionsDB** bevat 'n versameling IT-teoretiese terme en hul definisies. Die databasis bevat twee tabelle, naamlik **tblTopics** en **tblTerms**. 'n Program is nodig om 'n gebruiker in staat te stel om die inhoud van die databasis te bekyk en te redigeer.

Die databladsye wat aan die einde van die vraestel aangeheg is, verskaf inligting oor die ontwerp van die databasis en die inhoud daarvan.

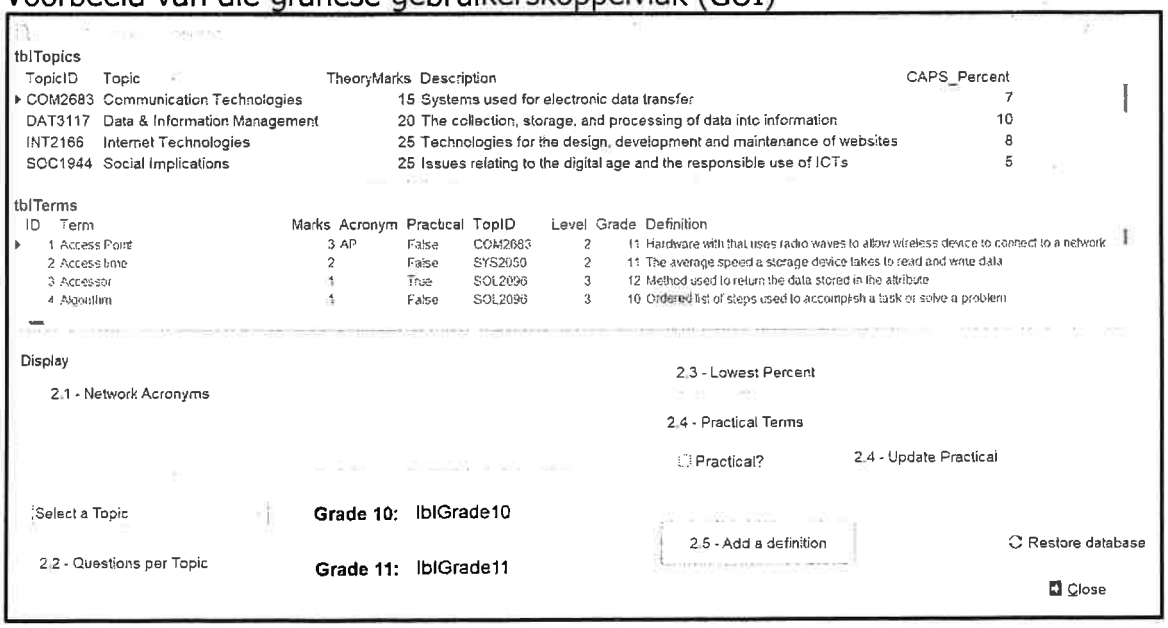
Doen die volgende:

- Maak die onvolledige projeklêer **frmQuestion2_P.dpr** in die gids **Vraag2** oop.
- Tik jou naam en van as kommentaar op die eerste reël van die **frmQuestion2_U.pas**-lêer.
- Die **Restore database** -knoppie word verskaf om die data in die databasis na die oorspronklike inhoud te herstel.
- Volg die instruksies om die kode vir elke vraag te voltooi soos hieronder beskryf.
- Moenie enige van die gegewe kode verander nie.
- Die databasis is reeds vir jou aan die GUI-komponente gekoppel.
- TWEE veranderlikes word as publieke veranderlikes verklaar, soos beskryf in die tabel hieronder.

Veranderlik	Datatipe	Beskrywing
tblTopics	TADOTable	Verwys na die databasistabel tblTopics
tblTerms	TADOTable	Verwys na die databasistabel tblTerms

2. In hierdie vraag mag slegs Delphi-programmeringskode gebruik word om VRAAG 2.1 tot VRAAG 2.5 te beantwoord

Voorbeeld van die grafiese gebruikerskoppelvlak (GUI)



2.1. **Button [2.1 – Network Acronyms]**

The user would like to see all the terms containing the word 'Network' that have an acronym, in **tblTerms**.

Write code to do the following:

- Find all the terms that contain the word 'Network' and have an acronym in the **Acronym** field. Ensure that your program is not case sensitive.
- Display the terms and acronyms in the rich edit **redQ2**.

Example of output:

```
LAN --> Local Area Network
NIC --> Network Interface Card
VPN --> Virtual Private Network
WAN --> Wide Area Network
WLAN --> Wireless Local Area Network
```

(12)

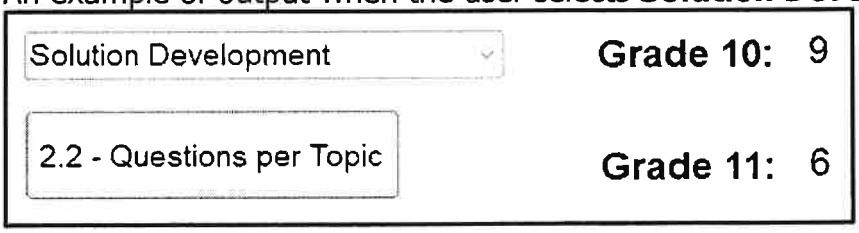
2.2. **Button [2.2 – Questions per Topic]**

Grade 11 learners would like to select a topic and see how many terms they have already learned in grade 10 and 11 from that topic.

Write code to do the following:

- Extract the topic selected from the combo box **cmbTopic**.
- Count how many terms in **tblTerms** belong to that topic and are taught in grade 10 and grade 11 respectively.
- Display the counts for each grade in the labels **lblGrade10** and **lblGrade11** respectively.

An example of output when the user selects **Solution Development**:



(18)

2.1. **Knoppie [2.1 – Network Acronyms]**

Die gebruiker wil graag al die terme sien wat die woord 'Network' bevat wat 'n akroniem het, in **tblTerms**.

Skryf kode om die volgende te doen:

- Soek al die terme wat die woord 'Network' bevat **en** 'n akroniem het in die **Acronym** -veld. Maak seker dat jou program nie hoofletter-sensitief is nie.
- Vertoon die terme en akronieme in die rich edit **redQ2**.

Voorbeeld van afvoer:

```
LAN --> Local Area Network
NIC --> Network Interface Card
VPN --> Virtual Private Network
WAN --> Wide Area Network
WLAN --> Wireless Local Area Network
```

(12)

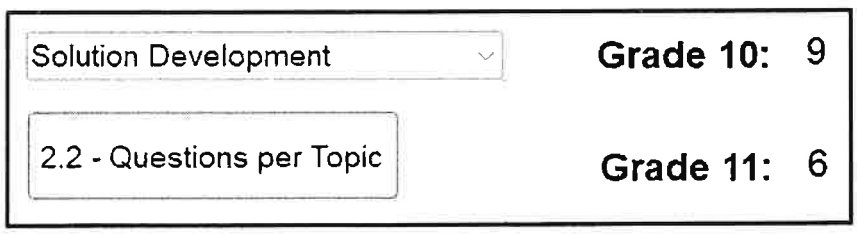
2.2. **Knoppie [2.2 – Questions per Topic]**

Graad 11-leerders wil graag 'n onderwerp kies en sien hoeveel terme hulle reeds in graad 10 en 11 van daardie onderwerp geleer het.

Skryf kode om die volgende te doen:

- Onttrek die onderwerp wat gekies is uit die kombinasieblokkie ("combo box") **cmbTopic**.
- Tel hoeveel terme in **tblTerms** tot daardie onderwerp behoort en onderskeidelik in graad 10 en graad 11 onderrig word.
- Vertoon die tellings vir elke graad in die byskrifte **lblGrade10** en **lblGrade11** onderskeidelik.

'n Voorbeeld van afvoer wanneer die gebruiker **Solution Development** kies:



(18)

2.3. Button [2.3 – Lowest Percent]

A learner wants to know which topics to focus on when studying for the theory exam. They want to find the topic with the lowest weighting in CAPS (the document that prescribes the curriculum) and store this information in a text file.

Write code to do the following:

- Find the topic and CAPS percent for the topic that has the lowest weighting in CAPS.
- Write the topic and the weighting with a suitable message to a new text file named '**Lowest.txt**'.
- Disable the button **btnQ2_3** to indicate that the file has been created.

NOTE: Clicking the **Restore database** button will enable **btnQ2_3**.

Example of the contents of the text file '**Lowest.txt**':

Social Implications is the lowest weighted at 5%

(10)

2.4. Button [2.4 – Update Practical]

A feature is needed to allow the user to change the value of the Practical field in **tblTerms**, for the active record, using the checkbox.

Write code to do the following:

- The user will select a record in the DBGrid for **tblTerms**.
- Update the Practical field for that record based on whether the checkbox **chkPractical** was selected.
- Display a suitable message, using a dialog box, to indicate to the user which term was changed.

An extract of output in the DBGrid **before** the button is clicked, with 'Algorithm' as active record:

▶	4 Algorithm	1	False	SOL2096

An extract of output in the DBGrid **after** the button is clicked with the checkbox selected with 'Algorithm' as active record:

▶	4 Algorithm	1	True	SOL2096

2.3. **Knoppie [2.3 – Lowest Percent]**

'n Leerder wil weet op watter onderwerpe moet daar gefokus word, wanneer hy vir die teorie-eksamen studeer. Hulle wil die onderwerp met die laagste gewig in KABV ("CAPS" - die dokument wat die kurrikulum voorskryf) vind en hierdie inligting in 'n tekslêer stoor.

Skryf kode om die volgende te doen:

- Vind die onderwerp en KABV persentasie vir die onderwerp wat die laagste gewig in KABV het.
- Skryf die onderwerp en die gewig met 'n geskikte boodskap na 'n nuwe tekslêer genaamd **'Lowest.txt'**.
- Deaktiveer die knoppie **btnQ2_3** om aan te dui dat die lêer geskep is.

LET WEL: Deur op die **Restore database**-knoppie te klik, sal **btnQ2_3** geaktiveer word.

Voorbeeld van die inhoud van die tekslêer **'Lowest.txt'**:

Social Implications is the lowest weighted at 5%

(10)

2.4. **Knoppie [2.4 – Update Practical]**

'n Eienskap is nodig om die gebruiker in staat te stel om die waarde van die Practical-veld in **tblTerms** te verander, vir die aktiewe rekord, deur die merkblokkie("check box") te gebruik.

Skryf kode om die volgende te doen:

- Die gebruiker sal 'n rekord in die DBGrid vir **tblTerms** kies.
- Indien die merkblokkie **chkPractical** gekies is, moet die Practical-veld opgedateer word.
- Vertoon 'n geskikte boodskap deur 'n dialoogkassie te gebruik, om aan die gebruiker aan te dui watter term verander is.

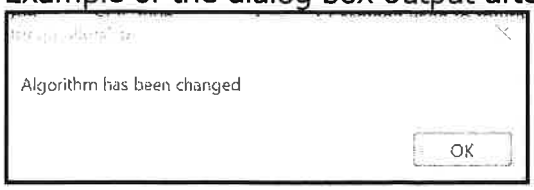
'n Uittreksel van afvoer in die DBGrid **voordat** die knoppie geklik word, met 'Algorithm' as aktiewe rekord:

▶	4 Algorithm	1	False	SOL2096

'n Uittreksel van afvoer in die DBGrid **nadat** die knoppie geklik is en met die merkblokkie gekies, met 'Algorithm' as aktiewe rekord:

▶	4 Algorithm	1	True	SOL2096

Example of the dialog box output after the record was updated:



(5)

2.5. **Button [2.5 - Add a definition]**

Write code to add a new term to the **tblTerms** table. The data for the new term is as follows:

Term :	Bit
Marks:	1
Practical :	False
TopID :	DAT3117
Definition :	Binary digit 0 or 1

The *TopID* and *Definition* have been added to the event as comments.

An extract of output once the record was added:

ID	Term	Marks	Acronym	Practical	TopID	Level	Grade	Definition
▶ 174	Bit	1		False	DAT3117			Binary digit 0 or 1

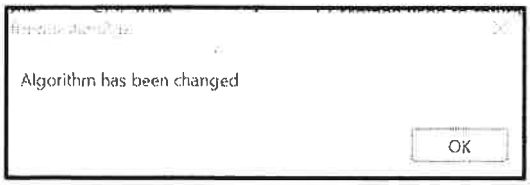
(5)

NOTE: You can add the term ONCE only. If you want to test your code by adding the term again, first restore the database to its original content before testing the code again.

- Enter your Name and Surname as a comment in the first line of the program file **frmQuestion2_U**.
- Save all the files ('File/Save All').
- Printouts of the code for the unit (**frmQuestion2_U.pas**) will be required.

TOTAL QUESTION 2: 50

Voorbeeld van die dialoogkassie-afvoer nadat die rekord opgedateer is:



(5)

2.5. **Knoppie [2.5 - Add a definition]**

Skryf kode om 'n nuwe term by die **tblTerms**-tabel te voeg. Die data vir die nuwe kwartaal is soos volg:

Term:	Bit
Marks:	1
Practical:	False
TopID:	DAT3117
Definition:	Binary digit 0 or 1

Die *TopID* en *Definition* is bygevoeg as kommentaar.

'n Uittreksel van afvoer sodra die rekord bygevoeg is:

ID	Term	Marks	Acronym	Practical	TopID	Level	Grade	Definition
▶ 174	Bit	1		False	DAT3117			Binary digit 0 or 1

(5)

LET WEL: Jy kan die term slegs EEN keer byvoeg. As jy jou kode wil toets deur die term weer by te voeg, herstel eers die databasis na sy oorspronklike inhoud voordat jy die kode weer toets.

- Tik jou Naam en Van as 'n opmerking in die eerste reël van die programlêer **frmQuestion2_U**.
- Stoor al die lêers ('File/Save All').

TOTAAL VRAAG 2: 50

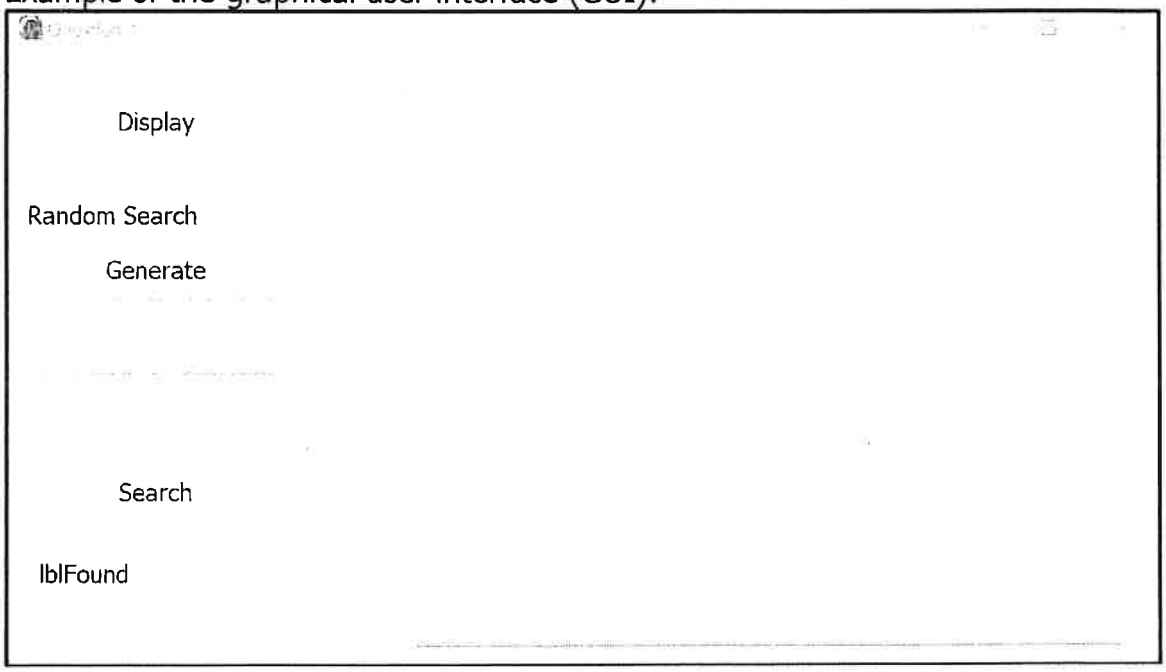
QUESTION 3: PROBLEM-SOLVING PROGRAMMING

Scenario: A program is needed to check whether a group of learners' answers to a Binary conversion question are correct. The user must also be able to search for the Binary numbers that are contained in a text file.

Do the following:

- Open the incomplete program in the folder **Question3**.
- Enter your name and surname as a comment on the first line of the **frmQuestion3_U.pas** file.
- Compile and execute the program. The program has no functionality currently.

Example of the graphical user interface (GUI):



The program contains the code shown below for the declaration of three parallel arrays called **arrName, arrBinary, arrAnswer** along with a counter variable called **iCount**:

- arrName: array[1..100] of String;
  //Stores the name and surname of the learners

- arrBinary : array [1..100] of String;
  //Stores the Binary number the learners need to convert

- arrAnswer : array[1..100] of Integer;
  //Stores the learners' decimal number answer to the Binary conversion

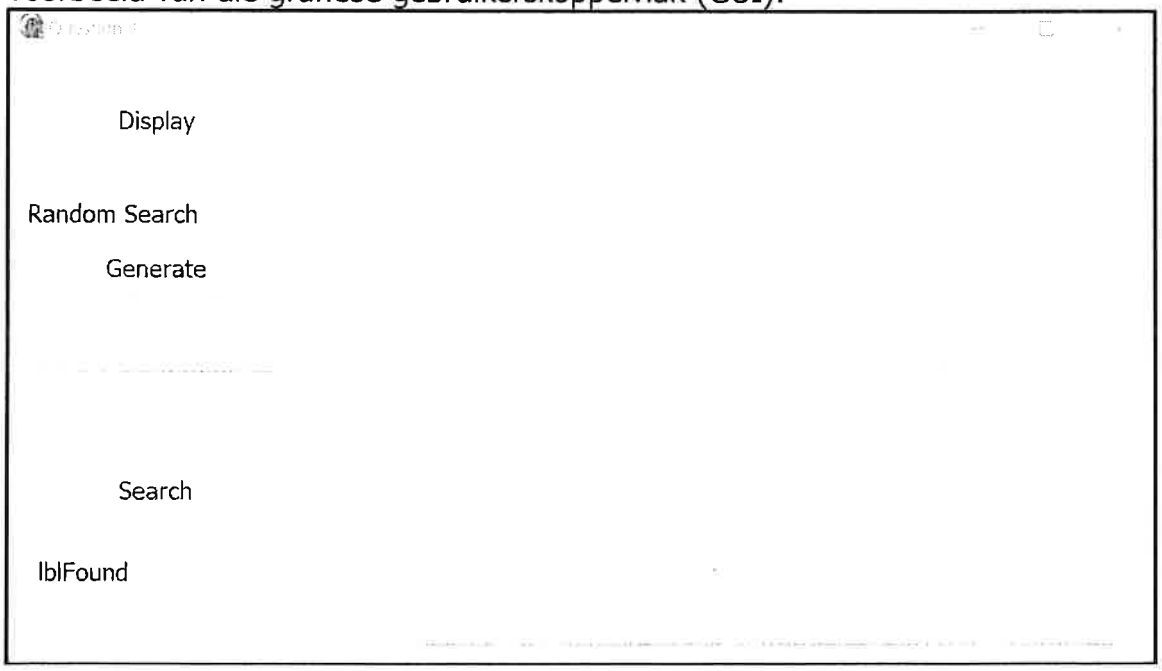
VRAAG 3: PROBLEEM-OPLOSSINGSPROGRAMMERING

Scenario: 'n Program is nodig om te kontroleer of 'n groep leerders se antwoorde op 'n Binêre omskakelingsvraag korrek is. Die gebruiker moet ook in staat wees om te soek na die Binêre nommers wat in 'n tekslêer vervat is.

Doen die volgende:

- Maak die onvolledige program oop in die gids **Vraag3**.
- Tik jou naam en van as kommentaar in die eerste reël van die **frmQuestion3_U.pas**-lêer.
- Stel die program saam en voer dit uit. Die program het tans geen funksionaliteit nie.

Voorbeeld van die grafiese gebruikerskoppelvlak (GUI):



Die program bevat die kode wat hieronder getoon word, vir die verklaring van drie parallelle skikkings genaamd **arrName**, **arrBinary**, **arrAnswer** saam met 'n teller-veranderlike genaamd **iCount**:

- arrName: array[1..100] of String;
  //Stoor die naam en van van die leerders

- arrBinary : array [1..100] of String;
  //Stoor die Binêre getal wat die leerders moet omskakel

- arrAnswer : array[1..100] of Integer;
  //Stoor die leerders se desimale antwoord by die Binêre omskakeling

A text file **Decimal.txt** is provided and contains data about the learners, Binary numbers to convert to decimal numbers and the learners' answers to the conversion in the following format:

<Name and surname>,<Binary Number>,<Decimal conversion>

An extract from the text file **Decimal.txt**:

```
Madison Henderson,1000010,66
Ndabicacile Mkhumbuzi,100110,38
Marc Mell,101100,44
Njabulo Ludonga,101111,46
```

Complete the code as described in QUESTION 3.1 to QUESTION 3.5 below:

3.1. Form Activate

Code has been provided to test if the file **Decimal.txt** exists and exit the event if it does not.

Write code to do the following:

- Open the file for reading.
- Make use of a loop to separate each line in the text file and store the data in the appropriate arrays provided.
- Use the counter to keep track of the number of items stored in the arrays.

(13)

3.2. Button [Display]

A **SetTabs** procedure has been provided to set tab stops and add headings to **redOutput**.

Code has provided to call the **SetTabs** procedure in the button.

Write code to do the following:

- Display the names and surnames, Binary numbers and the decimal answers neatly in columns in the rich edit **redOutput**.

An extract of output:

Name	Binary	Answer
Madison Henderson	1000010	66
Ndabicacile Mkhumbuzi	100110	38
Marc Mell	101100	44
Njabulo Ludonga	101111	46
Sakhiwo Sigcawu	1110100	118
Willem Gardiner	10000010	130
Bekithemba Hlabisa	10000001	129
Matthys Bekker	11000011	195

(4)

'n Tekslêer **Decimal.txt** word verskaf en bevat data oor die leerders, Binêre getalle wat omgeskakel moet word na desimale getalle en die leerders se antwoorde op die omskakeling in die volgende formaat:

<Naam en van>,<Binêre Getal>,<Desimale omskakeling>

'n Uittreksel uit die tekslêer **Decimal.txt**:

```
Madison Henderson,1000010,66
Ndabicacile Mkhumbuzi,100110,38
Marc Mell,101100,44
Njabulo Ludonga,101111,46
```

Voltooi die kode soos beskryf in VRAAG 3.1 tot VRAAG 3.5 hieronder:

3.1. **Form Activate**

Kode is verskaf om te toets of die lêer **Decimal.txt** bestaan en die gebeurtenis("event") te verlaat, as dit nie die geval is nie.

Skryf kode om die volgende te doen:

- Maak die lêer oop vir lees.
- Maak gebruik van 'n lus om elke reël in die tekslêer te skei en stoor die data in die toepaslike skikkings wat voorsien word.
- Gebruik die teller om tred te hou met die aantal items wat in die skikkings gestoor is.

(13)

3.2. **Knoppie [Display]**

'n **SetTabs**-prosedure is verskaf om tabel-stoppunte te stel en opskrifte by **redOutput** te voeg.

Kode is verskaf om die **SetTabs**-prosedure in die knoppie te roep.

Skryf kode om die volgende te doen:

- Vertoon die name en vanne, Binêre getalle en die desimale antwoorde netjies in kolomme in die rich edit **redOutput**.

'n Uittreksel van uitset:

Name	Binary	Answer
Madison Henderson	1000010	66
Ndabicacile Mkhumbuzi	100110	38
Marc Mell	101100	44
Njabulo Ludonga	101111	46
Sakhiwo Sigcawu	1110100	118
Willem Gardiner	10000010	130
Bekithemba Hlabisa	10000001	129
Matthys Bekker	11000011	195

(4)

3.3. **Button [Generate]**

The user can click the **Generate** button to generate a random Binary number and display it in the edit **edtBinary**.

Write code to do the following:

- Generate a random Binary number consisting of between 5 and 8 digits.
- The Binary number must start with a 1.
- Display the Binary number in the edit **edtBinary**.

Examples of output:

(6)

NOTE: Due to random numbers being generated your output may differ.

3.4. **Function ConvertToDecimal**

Create a function called **ConvertToDecimal** and complete the function to do the following:

- The function will receive a Binary number as a parameter and return the corresponding, converted Decimal number as an integer.
- Follow the Flowchart on the next page to write code to receive a Binary number as a string, convert it to a Decimal number and return it. (13)

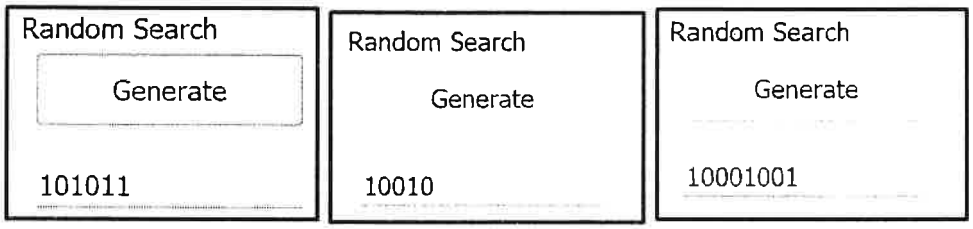
3.3. **Knoppie [Generate]**

Die gebruiker kan op die **Generate** -knoppie klik om 'n ewekansige Binêre nommer te genereer en dit in die edit **edtBinary** te vertoon.

Skryf kode om die volgende te doen:

- Genereer 'n ewekansige Binêre getal wat uit tussen 5 en 8 syfers bestaan.
- Die Binêre nommer moet met 'n 1 begin.
- Vertoon die Binêre nommer in die edit **edtBinary**.

Voorbeelde van uitset:



(6)

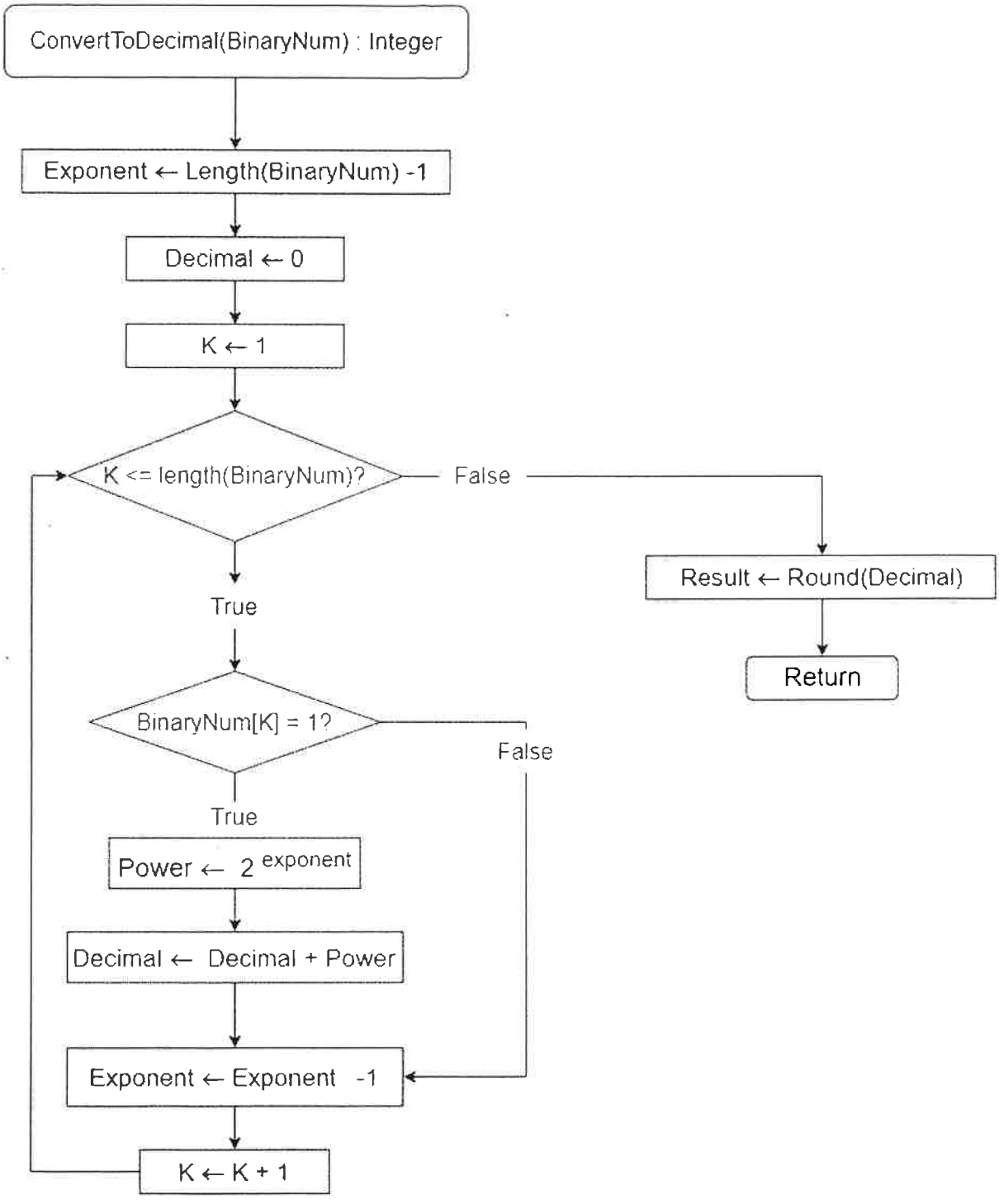
LET WEL: As gevolg van ewekansige getalle wat gegenereer word, kan jou afvoer verskil.

3.4. **Funksie ConvertToDecimal**

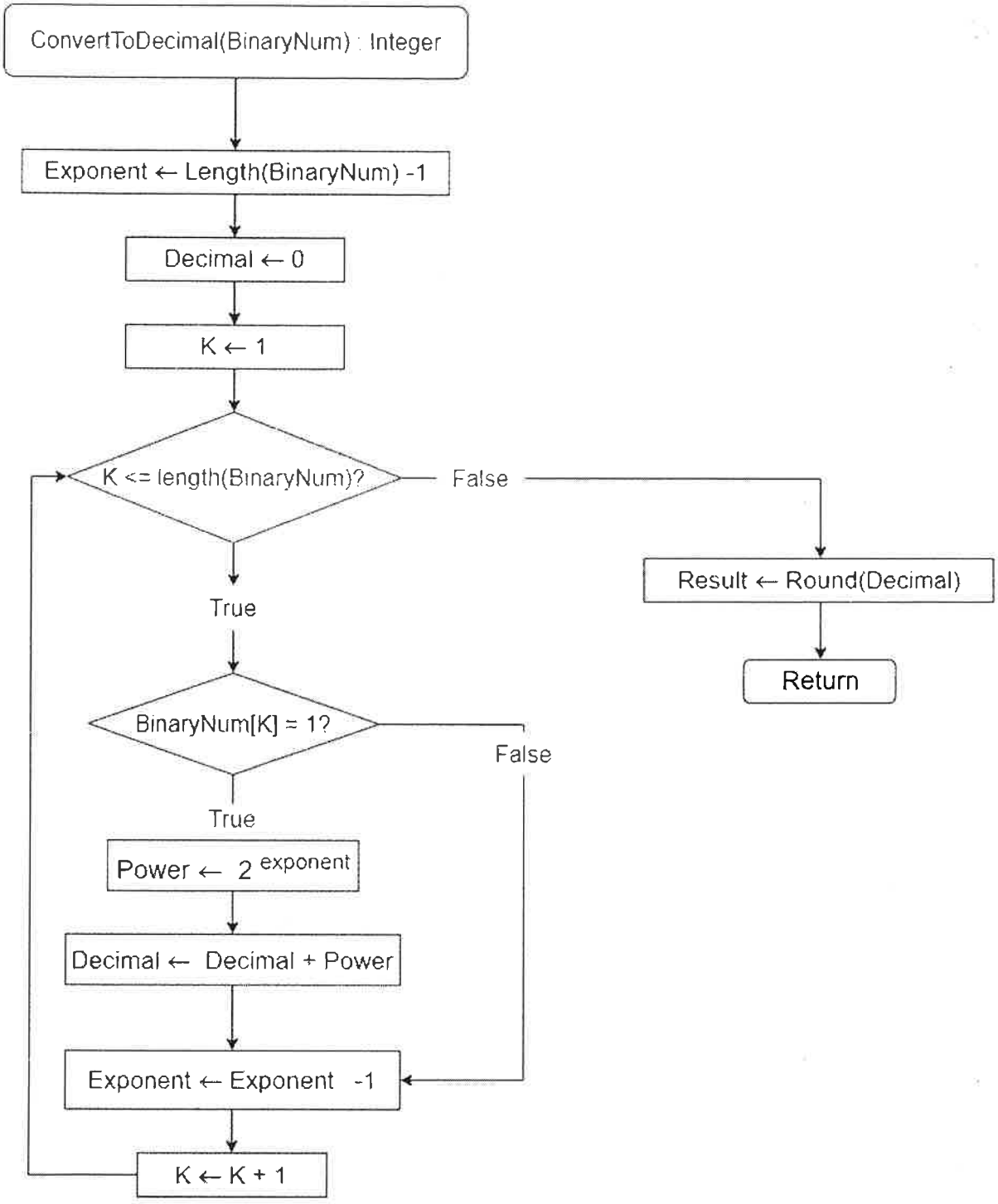
Skep 'n funksie genaamd **ConvertToDecimal** en voltooi die funksie om die volgende te doen:

- Die funksie sal 'n Binêre getal as 'n parameter ontvang en die ooreenstemmende, omgeskakelde Desimale getal as 'n heelgetal terugstuur.
- Volg die vloeidiagram op die volgende bladsy om kode te skryf om 'n Binêre getal as 'n string te ontvang, dit te omskep na 'n Desimale getal en dit dan terug te stuur. (13)

Flowchart for Function ConvertToDecimal



Vloeidiagram vir Funksie ConvertToDecimal



3.5. **Button [Search]**

The user can either use the randomly generated Binary number in **edtBinary** or they can enter their own Binary number in the edit to see the decimal conversion answer.

Code has been provided to add a heading in blue to the rich edit.

Write code to do the following:

- Store input from the user for the Binary number to search for.
- Call the **ConvertToDecimal** function and pass the Binary number as a parameter to the function.
- Display the Decimal number returned by the function in the label **lblAnswer**.
- Search through the array to find all Binary numbers in **arrBinary** that match the input Binary number.
- When a match is found, compare the Decimal number returned by the function to the Decimal number in **arrAnswer**.
    - If the numbers are the same, display the learner's name with the word <CORRECT>, in the rich edit **redOutput**.
    - If the numbers are not the same, display the learner's name with their answer.
- If no matching Binary number is found in the array, display a suitable message in a Dialog Box.

Example of output when **10001001** is entered or generated:

Output in **lblAnswer**:

Decimal number: 137

Output in **redOutput**

Answers tested Raymond Steel: 138 Mandlenkosi Ndlondlo: 138 Matanzima Soga: <CORRECT>

3.5. **Knoppie [Search]**

Die gebruiker kan óf die lukraak gegenereerde Binêre nommer in **edtBinary** gebruik óf hulle kan hul eie Binêre nommer in die edit invoeg om die desimale omskakelingsantwoord te sien.

Kode is verskaf om 'n opskrif in blou, in die rich edit te voeg.

Skryf kode om die volgende te doen:

- Stoor die toevoer van die gebruiker om vir die Binêre nommer te soek.
- Roep die **ConvertToDecimal**-funksie en gee die Binêre getal as 'n parameter vir die funksie.
- Vertoon die desimale getal wat deur die funksie teruggestuur word, in die label **lblAnswer**.
- Soek deur die skikking om alle Binêre getalle in **arrBinary** te vind wat ooreenstem met die toevoer.
- Wanneer daar wel een gevind is, vergelyk die Desimale getal wat deur die funksie teruggestuur word met die Desimale getal in **arrAnswer**.
    - As die getalle dieselfde is, vertoon die leerder se naam met die woord <CORRECT>, in die rich edit **redOutput**.
    - As die getalle nie dieselfde is nie, vertoon die leerder se naam saam met hul antwoord.
- As geen ooreenstemmende Binêre nommer in die skikking gevind word nie, vertoon 'n geskikte boodskap in 'n dialoogkassie.

Voorbeeld van afvoer wanneer **10001001** ingevoer of gegenereer word:

Afvoer in **lblAnswer**:

Decimal number: 137

Afvoer in **redOutput**

Answers tested Raymond Steel: 138 Mandlenkosi Ndlondlo: 138 Matanzima Soga: <CORRECT>

Example of output when **10101010** is entered or generated:

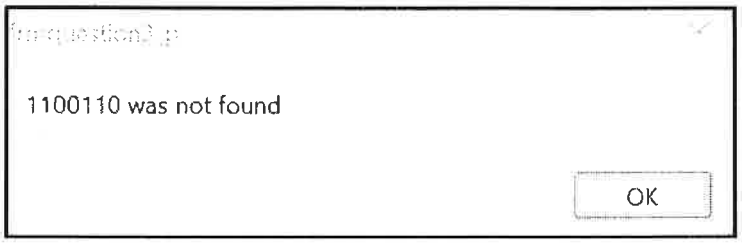
Output in **lblAnswer**:

Decimal number: 170

Output in **redOutput**:

Answers tested Zimkhitha Vetyeka: <CORRECT> Johan Bakkes: 178 Craig Gold: 177

Example of output when **1100110** is searched for:



(14)

- Enter your Name and Surname as a comment in the first line of the program file **Question3_U**.
- Save all the files ('File/Save All').
- Printouts of the code for the unit (**frmQuestion3_U**) will be required.

TOTAL QUESTION 3: 50

GRAND TOTAL: 150

Voorbeeld van afvoer wanneer **10101010** ingevoer of gegenereer word:

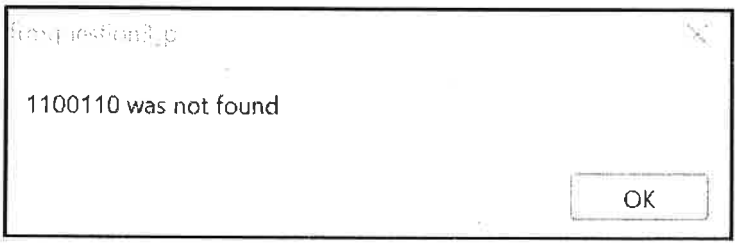
Afvoer in **lblAnswer**:

Decimal number: 170

Afvoer in **redOutput**:

Answers tested Zimkhitha Vetyeka: <CORRECT> Johan Bakkes: 178 Craig Gold: 177

Voorbeeld van afvoer wanneer **1100110** gesoek word:



(14)

- Tik jou Naam en Van as 'n opmerking in die eerste reël van die programlêer Vraag3_U.
- Stoor al die lêers ('File/Save All').

TOTAAL VRAAG 3: 50

GROOTTOTAAL: 150

INFORMATION TECHNOLOGY P1

QUESTION 2: DATABASE INFORMATION

DESIGN OF DATABASE TABLES:

Table: **tblTopics**

This table contains data about the different IT theory topics.

Field name	Data type	Description
TopicID(PK)	Text	A unique value assigned to each topic
Topic	Text	The name of the topic
TheoryMarks	Number	The approximate number of marks this topic gets in theory exams
Description	Text	Description of what the topic covers
CAPS_Percent	Number	The percentage of how much this topic counts in the curriculum

Example of data from **tblTopics**:

TopicID	Topic	TheoryMarks	Description	CAPS_Percent
COM2683	Communication Technologies	15	Systems used for electronic data transfer	7
DAT3117	Data & Information Management	20	The collection, storage, and processing of data into information	10
INT2166	Internet Technologies	25	Technologies for the design, development and maintenance of websites	8
SOC1944	Social Implications	25	Issues relating to the digital age and the responsible use of ICTs	5
SOL2096	Solution Development	30	Algorithms and creating a software solution according to set requireme	60
SYS2050	Systems Technologies	35	Hardware and software for the computer to perform computing functions	10

INLIGTINGSTEGNOLOGIE V1

VRAAG 2: DATABASISINLIGTING

ONTWERP VAN DATABASIS-TABELS:

Tabel: **tblTopics**

Hierdie tabel bevat data oor die verskillende IT-teorie-onderwerpe.

Veld naam	Datatipe	Beskrywing
TopicID(PK)	Text	'n Unieke waarde wat aan elke onderwerp toegeken word
Topic	Text	Die naam van die onderwerp
TheoryMarks	Number	Die benaderde aantal punte wat hierdie onderwerp in teorie-eksamens kry
Description	Text	Beskrywing van wat die onderwerp dek
CAPS_Percent	Number	Die persentasie van hoeveel hierdie onderwerp in die kurrikulum tel

Voorbeeld van data van **tblTopics**:

TopicID	Topic	TheoryMarks	Description	CAPS_Percent
COM2683	Communication Technologies		15 Systems used for electronic data transfer	7
DAT3117	Data & Information Management		20 The collection, storage, and processing of data into information	10
INT2166	Internet Technologies		25 Technologies for the design, development and maintenance of websites	8
SOC1944	Social Implications		25 Issues relating to the digital age and the responsible use of ICTs	5
SOL2096	Solution Development		30 Algorithms and creating a software solution according to set requireme	65
SYS2050	Systems Technologies		35 Hardware and software for the computer to perform computing functions	10

Table: **tblTerms**

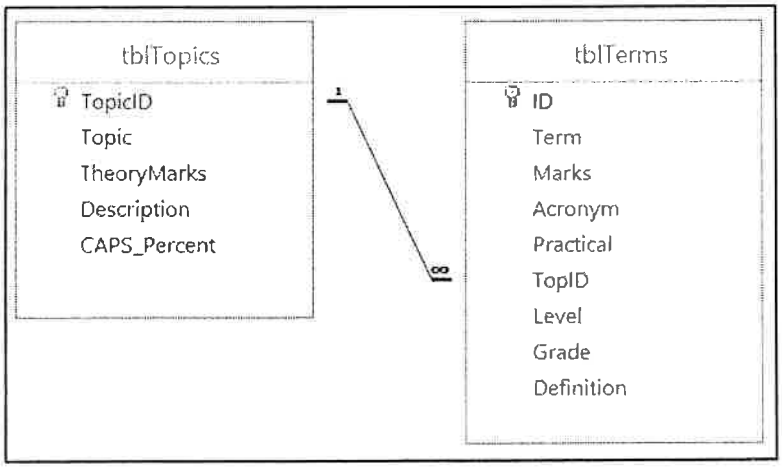
This table contains data about IT theory terms

Field name	Data type	Description
ID(PK)	AutoNumber	A unique number assigned to each term
Term	Text	The theory term
Marks	Number	The number of marks this term's definition counts in an exam
Acronym	Text	The term's acronym, if it has one
Practical	Yes/No	Is this a practical programming term
TopID(FK)	Text	An ID that connects the term to the topic it belongs to
Level	Number	The term's difficulty level between 1 and 3, 3 being the most difficult
Grade	Number	The grade in which the term is covered, 10, 11 or 12
Definition	Text	The definition of the theory term

Example of data of the first ten records:

ID	Term	Marks	Acronym	Practical	TopID	Level	Grade	Definition
1	Access Point	3	AP	☐	COM2683	2	11	Hardware with that uses radio waves to allow wireless device to connect to a network
2	Access time	2		☐	SYS2050	2	11	The average speed a storage device takes to read and write data
3	Accessor	1		☑	SOL2096	3	12	Method used to return the data stored in the attribute
4	Algorithm	1		☐	SOL2096	3	10	Ordered list of steps used to accomplish a task or solve a problem
5	Anti-Virus	1		☐	SYS2050	1	10	Software that prevents your computer from being affected by computer viruses
6	Archive	2		☐	SYS2050	2	10	Less frequently used files are compressed and stored on another storage device
7	Array	1		☑	SOL2096	2	11	Data structure that stores a set values under the same type
8	Artificial intelligence	2	AI	☐	SOC1944	3	11	Simulation of human decision making in machines programmed to think and react like
9	Assembler	1		☐	SYS2050	1	11	Program that translates assembler code to machine code
10	Attenuation	2		☐	COM2683	2	11	The loss of signal strength in a network medium over a distance

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



Tabel: **tblTerms**

Hierdie tabel bevat data oor IT-teoretiese terme

Veld naam	Datatipe	Beskrywing
ID(PK)	AutoNumber	'n Unieke nommer word aan elke kwartaal toegeken
Term	Text	Die teorie term
Marks	Number	Die aantal punte wat hierdie term se definisie tel in 'n eksamen
Acronym	Text	Die term se akroniem, as dit een het
Practical	Yes/No	Is dit 'n praktiese programmeringstermyn
TopID(FK)	Text	'n ID wat die term verbind met die onderwerp waaraan dit behoort
Level	Number	Die term se moeilikheidsgraad tussen 1 en 3, 3 is die moeilikste
Grade	Number	Die graad waarin die kwartaal gedek word, 10, 11 of 12
Definition	Text	Die definisie van die teorieterm

Voorbeeld van data van die eerste tien rekords:

ID	Term	Marks	Acronym	Practical	TopID	Level	Grade	Definition
1	Access Point	3	AP	☐	COM2683	2	11	Hardware with that uses radio waves to allow wireless device to connect to a network
2	Access time	2		☐	SYS2050	2	11	The average speed a storage device takes to read and write data
3	Accessor	1		☑	SOL2096	3	12	Method used to return the data stored in the attribute
4	Algorithm	1		☐	SOL2096	3	10	Ordered list of steps used to accomplish a task or solve a problem
5	Anti-Virus	1		☐	SYS2050	1	10	Software that prevents your computer from being affected by computer viruses
6	Archive	2		☐	SYS2050	2	10	Less frequently used files are compressed and stored on another storage device
7	Array	1		☑	SOL2096	2	11	Data structure that stores a set values under the same type
8	Artificial intelligence	2	AI	☐	SOC1944	3	11	Simulation of human decision making in machines programmed to think and react like
9	Assembler	1		☐	SYS2050	1	11	Program that translates assembler code to machine code
10	Attenuation	2		☐	COM2683	2	11	The loss of signal strength in a network medium over a distance

Die volgende een-tot-baie-verhouding met verwysingsintegriteit("referential integrity") bestaan tussen die twee tabelle in die databasis: