

Vertroulik



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2024

PUNTE: 150

TYD: 3 uur

**Hierdie vraestel bestaan uit 25 bladsye, 2 bladsye met data,
2 bladsye vir beplanning en 'n aparte inligtingsblad.**

INSTRUKSIES EN INLIGTING

1. Hierdie vraestel is in VIER afdelings ingedeel. Kandidate moet AL die vrae in AL VIER afdelings beantwoord.
2. Twee leë bladsye wat vir beplanningsdoeleindes gebruik kan word, is aan die einde van die vraestel ingesluit.
3. Die duur van hierdie eksamen is drie uur. As gevolg van 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.
4. Hierdie vraestel is opgestel met programmeringsterme wat spesifiek is tot Delphi as programmeringstaal. Die Delphi-programmeringstaal moet gebruik word om die vrae te beantwoord.
5. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal volgens die voorgeskrewe vereistes toegeken word.
6. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, indien die vraag nie vir datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
7. Jou programme moet op só 'n manier gekodeer word dat dit met enige data sal werk en nie net met die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel verskyn nie.
8. Roetines, soos soek, sorteer en seleksie, moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van die Delphi-programmeringstaal vir enige van hierdie roetines gebruik NIE.
9. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture verskaf word.
10. Jy moet jou werk gereeld stoor op die disket/CD/DVD/geheuestokkie wat aan jou gegee is of op die skyfspasie wat vir hierdie eksamensessie aan jou toegeken is.
11. Maak seker dat jou eksamennommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
12. Indien dit vereis word, druk die programmeringskode van al die programme/klasse wat jy voltooi het. Jou eksamennommer moet op alle drukstukke voorkom. Jy sal ná die eksamensessie 'n halfuur tyd vir drukwerk gegee word.
13. Aan die einde van hierdie eksamensessie moet jy 'n disket/CD/DVD/geheuestokkie inlewer met al jou werk daarop gestoor OF jy moet seker maak dat al jou werk op die skyfspasie gestoor is wat vir hierdie eksamensessie aan jou toegeken is. Maak seker dat al die lêers gelees kan word.

14. Die lêers wat jy benodig om hierdie vraestel te voltooi, is aan jou gegee op 'n disket/CD/DVD/geheuestokkie of op die skyfspasie wat aan jou toegeken is. Die lêers word in die vorm van wagwoordbeskermdes uitvoerbare lêers verskaf. ...

Doen die volgende:

- Dubbelklik op die volgende uitvoerbare lêer wat met 'n wagwoord beskerm is: **DataNOV2024.exe**
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **#RUN4@New**

Nadat dit onttrek ('extracted') is, sal die volgende lys lêers in die lêergids ('folder') **DataNOV2024** beskikbaar wees:

Question 1:

Houses.txt
Question1_P.dpr
Question1_P.dproj
Question1_P.res
Question1_U.dfm
Question1_U.pas

Question 2:

ConnectDB_U.pas
MarathonsDB - Copy.mdb
MarathonsDB.mdb
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question 3:

MRecord_U.pas
Question3_P.dpr
Question3_P.dproj
Question3_P.res
Question3_U.dfm
Question3_U.pas

Question 4:

Question4_P.dpr
Question4_P.dproj
Question4_P.res
Question4_U.dfm
Question4_U.pas

AFDELING A**VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

Doen die volgende:

- Maak die onvolledige program in die **Question 1**-lêergids oop.
- Sleutel jou eksamenommer as kommentaar in die eerste reël van die **Question1_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')):

The screenshot shows a GUI application window titled "General programming skills". It contains five distinct sections for user interaction:

- 1.1**: Contains two buttons: "1.1.1 - Random" next to an empty text input field, and "1.1.2 - Round up" next to a spin box containing the number "0".
- 1.2**: Displays the mathematical formula $A = \pi r(r + \sqrt{h^2 + r^2})$ and a diagram of a cone with radius r and height h . Below the diagram are two input fields: "r = 5" and "h = 10". A button labeled "1.2 - Surface area" is positioned below these fields.
- 1.3**: Features a button labeled "1.3 - Read file" above a large, empty text area.
- 1.4**: Includes a "Name:" label with a dropdown menu, a "Paid:" label with an unchecked checkbox, and a button labeled "1.4 - Add name". Below this are two columns: "Paid" and "Not paid", each with an empty text input field.
- 1.5**: Shows a text input field containing "Lilly Johnson", a button labeled "1.5 - Replace", and another empty text input field below the button.

- Voltooi die kode vir elke afdeling van VRAAG 1, soos wat beskryf is in VRAAG 1.1 tot VRAAG 1.5 op die volgende bladsy.

1.1 Skryf kode vir knoppies **btnQ1_1_1** en **btnQ1_1_2** soos hieronder gespesifiseer word.

1.1.1 **Knoppie [1.1.1 - Random]**

Skryf kode om die volgende te doen:

- Genereer 'n ewekansige getal van 5 tot 10, beide getalle ingesluit.
- Vertoon die ewekansige getal in die **edtQ1_1_1**-redigeerblokkie ('edit box').

Voorbeeld van afvoer indien die getal 5 ewekansig gegeneer is:

LET WEL: Die afvoer kan verskil as gevolg van die aard van ewekansige getalle.

(4)

1.1.2 **Knoppie [1.1.2 - Round up]**

Jy is van 'n konstante **NUMBER = 5.23247** voorsien.

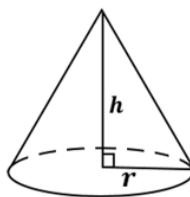
Skryf kode wat 'n wiskundige funksie gebruik om die waarde van die konstante te vertoon, afgerond tot die naaste heelgetal in die spinredigeerder ('spin edit') **spnQ1_1_2**.

Voorbeeld van afvoer:

(3)

1.2 **Knoppie [1.2 - Surface area]**

'n Reghoekige soliede keël word in die prentjie hieronder getoon.



Die buite-oppervlakte van die keël word bereken deur die volgende formule te gebruik:

$$A = \pi r(r + \sqrt{h^2 + r^2})$$

Skryf kode om die volgende te doen:

- Onttrek die waardes van **r** en **h** uit die onderskeie redigeerblokkies **edtQ1_2_r** en **edtQ1_2_h** wat voorsien is.
- Gebruik die waardes van **r** en **h** en geskikte wiskundige funksies om die buite-oppervlakte van die keël wat deur **A** verteenwoordig word, te bereken.
- Vertoon die waarde van **A** in die etiket ('label') **lblQ1_2**, afgerond tot TWEE desimale plekke.

Voorbeeld van toevoer en afvoer waar $r = 5$ en $h = 10$:

$r =$ <input type="text" value="5"/>	$h =$ <input type="text" value="10"/>
1.2 - Surface area	
254.16	

Voorbeeld van toevoer en afvoer waar $r = 7.9$ en $h = 22.4$:

$r =$ <input type="text" value="7.9"/>	$h =$ <input type="text" value="22.4"/>
1.2 - Surface area	
785.56	

(8)

1.3 Knoppie [1.3 - Read file]

'n Tekslêer met die naam **Houses.txt** bevat die straatadres en die getal slaapkamers vir elke huis in 'n voorstedelike gebied.

Die formaat van die inhoud van die tekslêer is soos volg:

```
<straatadres>
<getal slaapkamers>
```

Voorbeeld van die eerste vier reëls van die **Houses.txt**-tekstlêer:

```
61 Loop street
4
83 Church street
2
```

Kode is voorsien vir die verklaring van die tekstlêerveranderlike **tFile**.

Skryf kode om die volgende te doen:

- Ken die tekstlêer **Houses.txt** aan die veranderlike **tFile** toe.
- Maak seker dat lees uit die tekstlêer vanaf die eerste reël in die tekstlêer begin.
- Lees met 'n lus deur die tekstlêer en stoor die straatadres en getal slaapkamers in verskillende veranderlikes.
- Vertoon die straatadres en die getal slaapkamers in die redigeerblokkie ('rich edit') **redQ1_3** in die formaat hieronder.

```
<straatadres> - <getal slaapkamers> -bedroom house
```

Voorbeeld van afvoer:

61 Loop street - 4-bedroom house
83 Church street - 2-bedroom house
17 School street - 3-bedroom house

(9)

1.4 Knoppie [1.4 - Add name]

Leerders van 'n skool gaan op 'n uitstappie. Twee 'rich edit'-komponente **redQ1_4_P** en **redQ1_4_NP** is voorsien om onderskeidelik die name van die leerders te vertoon wat hulle fooie betaal het en dié wat nie hulle fooie betaal het nie.

Die kombinasieblokkie ('combo box') **cmbQ1_4** bevat die name van al die leerders wat op die uitstappie gaan.

Skryf kode om die volgende te doen:

- Onttrek die naam wat uit die kombinasieblokkie ('combo box') **cmbQ1_4** geselekteer is.
- Indien die 'Paid'-kontroleblokkie ('check box') **chbQ1_4** gemerk is, voeg die naam van die leerder by die **redQ1_4_P**-komponent, andersins moet die naam by die **redQ1_4_NP**-komponent gevoeg word.
- Verwyder die naam wat geselekteer is uit die kombinasieblokkie **cmbQ1_4**.

Voorbeeld van toevoer en afvoer indien Bianca geselekteer is en haar uitstappiefooie reeds betaal is:

Name: <input type="text" value="Bianca"/>	<input checked="" type="checkbox"/> Paid:
---	---

Paid	Not paid
<input type="text" value="Bianca"/>	<input type="text"/>

Voorbeeld van toevoer en afvoer indien Godfrey geselekteer is en sy uitstappiefooie nog nie betaal is nie:

Name: <input type="text" value="Godfrey"/>	<input type="checkbox"/> Paid:
--	--------------------------------

Paid	Not paid
<input type="text" value="Bianca"/>	<input type="text" value="Godfrey"/>

Voorbeeld van die inhoud van **cmbQ1_4** nadat Bianca en Godfrey geselekteer is en hulle name uit die kombinasieblokkie verwyder is:



(7)

1.5 Knoppie [1.5 - Replace]

'n Wagwoord moet vir elke leerder wat die uitstappie bywoon, saamgestel word.

Die gebruiker moet die naam en van van die leerder in die redigeerblokkie ('edit box') **edtQ1_5** insleutel.

Die volgende kode is voorsien:

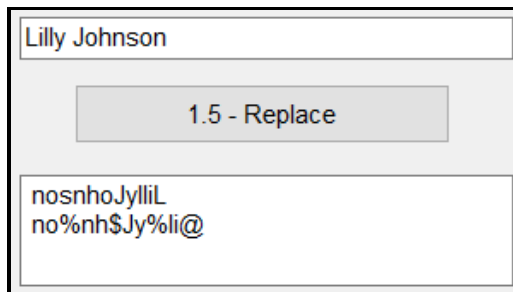
- Die inhoud van die redigeerblokkie **edtQ1_5** is in die veranderlike **sNameSurname** gestoor.
- 'n String, **sCharacters**, wat ses spesiale karakters bevat

```
sCharacters := '@#$$%^&';
```

Skryf kode om die volgende te doen om die wagwoord te skep:

- Keer die volgorde van die karakters in die string **sNameSurname** om en verwyder alle spasies uit die string.
- Vertoon die omgekeerde string, sonder spasies, in die **memQ1_5**-afvoerarea.
- Vervang elke derde karakter in die omgekeerde string met 'n spesiale karakter wat ewekansig uit die gegewe **sCharacter**-string verkry word.
- Vertoon die wagwoord in die **memQ1_5**-afvoerarea.

Voorbeeld van toevoer en afvoer:



(9)

- Sleutel jou eksamenommer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING A:

40

AFDELING B**VRAAG 2: DATABASISPROGRAMMERING**

Die organiseerders van atletiekbyeenkomste benodig 'n toepassing ('app') wat aan hardlooptoesiaste inligting oor marathons en waar dit plaasvind, voorsien.

'n Databasis met die naam **MarathonsDB.mdb**, wat inligting bevat oor marathonname en die stede waar dit plaasvind, is ontwikkel. Dit stel gebruikers in staat om marathons in die stede van hulle voorkeur te vind.

Die databasis bevat twee tabelle met die name **tblLocations** en **tblMarathons**.

LET WEL: Die bladsye met data aan die einde van die vraestel verskaf inligting oor die ontwerp van die **MarathonsDB.mdb**-databasis en die inhoud daarvan.

Doen die volgende:

- Maak die onvolledige projeklêer met die naam **Question2_P.dpr** in die **Question 2**-lêergids oop.
- Sleutel jou eksamennummer as kommentaar in die eerste reël van die **Question2_U.pas**-eenheidlêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie. Die inhoud van die tabelle word vertoon, soos wat hieronder op die seleksie van die oortjieblad ('tab sheet') **2.2 - Delphi code** getoon word.

LocationID	City	Province	Altitude	Population
1	Cape Town	Western Cape	42	4618000
2	Durban	KwaZulu-Natal	8	3721000
3	Bloemfontein	Free State	1395	598000
4	Pretoria	Gauteng	1339	2473000

MarathonID	MarathonName	MarathonDate	Distance	PrizeMoney	Organiser	LocationID
1	East London Marathon	2024/09/12	28.4	R50 000.00	Running South Africa	15
2	Joburg Run	2024/10/26	55.2	R12 000.00	Active Events	6
3	Cape Town Challenge	2025/01/18	40.6	R20 000.00	Runners United	1
4	Durban Ultra Marathon	2025/02/19	45.8	R10 000.00	Endurance Sports SA	2

- Volg die instruksies hieronder om die kode vir elke afdeling wat in VRAAG 2.1 en VRAAG 2.2 beskryf word, te voltooi.
- Gebruik SQL-stellings om VRAAG 2.1 te beantwoord en Delphi-kode om VRAAG 2.2 te beantwoord.

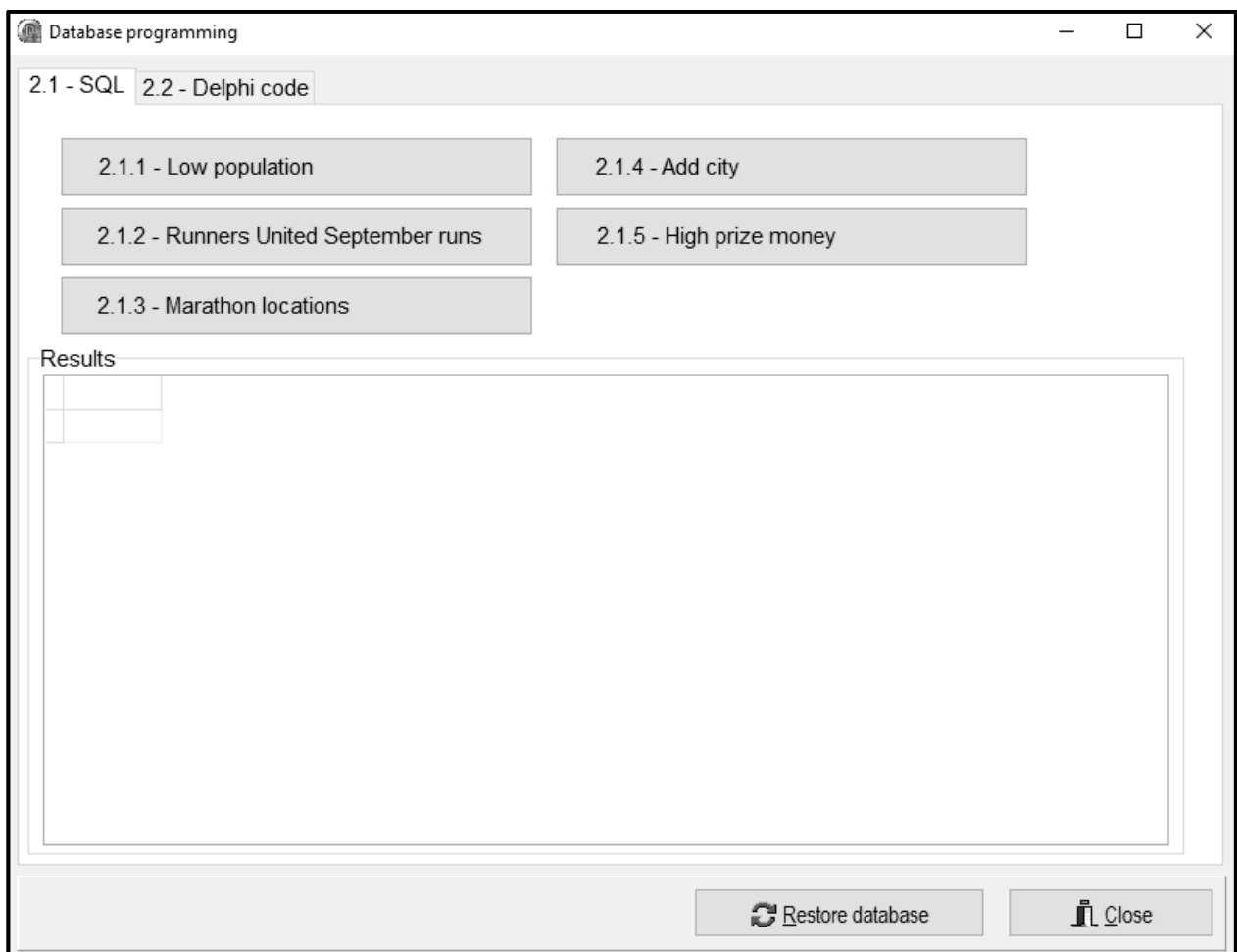
LET WEL:

- Die 'Restore database'-knoppie word voorsien om die data wat in die databasis voorkom, na die oorspronklike inhoud te herstel.
- Die inhoud van die databasis is met 'n wagwoord beskerm, m.a.w. jy sal NIE toegang tot die inhoud van die databasis kan verkry deur Microsoft Access te gebruik NIE.
- Kode word voorsien om die GGK('GUI')-komponente met die databasis te koppel. MOENIE enige van die kode wat voorsien is, verander NIE.
- TWEE veranderlikes is as publieke veranderlikes verklaar, soos in die tabel hieronder beskryf word.

Veranderlike	Datatype	Beskrywing
tblLocations	TADOTable	Verwys na die tabel tblLocations
tblMarathons	TADOTable	Verwys na die tabel tblMarathons

2.1 Oortjieblad ('Tab sheet') [2.1 - SQL]

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK) vir VRAAG 2.1:



LET WEL:

- Gebruik SLEGS SQL-stellings om VRAAG 2.1.1 tot VRAAG 2.1.5 te beantwoord.
- Kode om die SQL-stellings uit te voer en die resultate van die navrae te vertoon, is voorsien. Die SQL-stellings wat aan die veranderlikes **sSQL1**, **sSQL2**, **sSQL3**, **sSQL4** en **sSQL5** toegeken is, is onvolledig.

Voltooi die SQL-stellings om die take wat in VRAAG 2.1.1 tot VRAAG 2.1.5 hieronder beskryf word, uit te voer.

2.1.1 Knoppie [2.1.1 - Low population]

Vertoon al die velde van die rekords in die **tblLocations**-tabel met 'n **'population' (bevolking)** van minder as 200 000 mense.

Voorbeeld van afvoer:

LocationID	City	Province	Altitude	Population
9	Mbombela	Mpumalanga	1653	110159
11	Potchefstroom	North West	1127	67372
12	Kroonstad	Free State	1532	107514
17	Upington	Northern Cape	2742	87301

(3)

2.1.2 Knoppie [2.1.2 - Runners United September runs]

Vertoon die **MarathonID**, **MarathonDate** en **Distance** van al die marathons wat deur Runners United georganiseer is en wat in September plaasvind.

Voorbeeld van afvoer:

MarathonID	MarathonDate	Distance
13	2025/09/14	22.7
26	2025/09/27	70.2

(4)

2.1.3 Knoppie [2.1.3 - Marathon locations]

Die plek waar elke marathon plaasvind, word deur die organiseerders benodig.

Las die **City**-veld en die eerste drie letters van die **Province**-veld van al die plekke in die **tblLocations**-tabel aan mekaar in 'n nuwe veld met die naam **Location**.

Die inhoud van die nuwe **Location**-veld moet in die volgende formaat wees:

<Stad> - <Eerste drie letters van provinsie>

Voorbeeld van afvoer van die eerste vyf rekords:

Location
Polokwane - Lim
Mbombela - Mpu
eMalahleni - Mpu
Potchefstroom - Nor
Kroonstad - Fre

(4)

2.1.4 **Knoppie [2.1.4 - Add city]**

'n Stad moet in die **tblLocations**-tabel bygevoeg word. Die besonderhede is in die tabel hieronder voorsien.

Veld	Waarde
LocationID	19
City	Welkom
Province	Free State
Altitude	1198
Population	423016

Voorbeeld van afvoer van die laaste vyf rekords nadat die stad bygevoeg is:

15	East London	Eastern Cape	18	394034
16	Kimberly	Northern Cape	1116	284509
17	Upington	Northern Cape	2742	87301
18	Klerksdorp	North West	1297	226215
19	Welkom	Free State	1198	423016

(4)

2.1.5 **Knoppie [2.1.5 - High prize money]**

Vertoon die stad, die getal marathons en die totale prysgeld van al die marathons wat in die stad gehou word, slegs as die totale prysgeld meer as R50 000 is.

Die nuwe veld vir die getal marathons is **NumMarathons** en die nuwe veld vir die totale prysgeld is **Total Prize Money**.

Voorbeeld van afvoer:

City	NumMarathons	Total Prize Money
East London	2	64500
Mbombela	3	68500

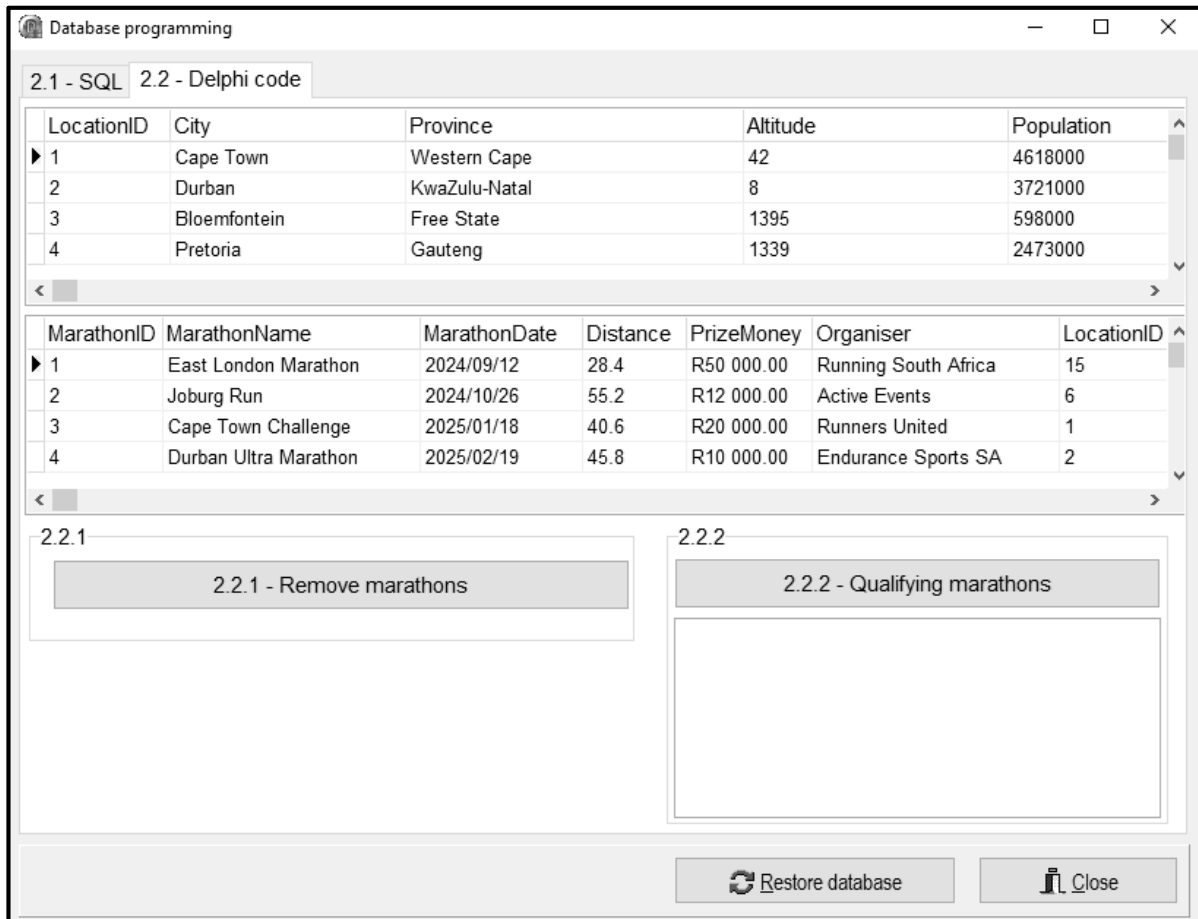
(7)

2.2 Oortjieblad ('Tab sheet') [2.2 - Delphi code]

LET WEL:

- Gebruik SLEGS Delphi-programmeringskode om VRAAG 2.2.1 en VRAAG 2.2.2 te beantwoord.
- GEEN punte sal vir SQL-stellings in VRAAG 2.2 toegeken word nie.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')) vir VRAAG 2.2:



2.2.1 Knoppie [2.2.1 - Remove marathons]

Sommige organiseerders het van die marathons onttrek en die rekords van hierdie marathons moet verwyder word aangesien hulle nie meer gaan plaasvind nie.

Die organiseerder se naam moet deur die gebruiker ingesleutel word.

Kode is voorsien om die naam van die organiseerder te onttrek ('extract') en te stoor deur 'n toevoerdialoogblokkie ('input dialog box') te gebruik. Die naam van die organiseerder word in die veranderlike **sOrganiser** gestoor.

Skryf kode om die rekords uit die **tblMarathons**-tabel te verwyder, waar die naam van die organiseerder dieselfde is as die naam van die organiseerder wat ingesleutel is.

Voorbeeld van die eerste vier rekords in die **tblMarathons**-tabel voordat die organiseerder, Active Events, uitgegee is:

MarathonID	MarathonName	MarathonDate	Distance	PrizeMoney	Organiser	LocationID
1	East London Marathon	2024/09/12	28.4	R50 000.00	Running South Africa	15
2	Joburg Run	2024/10/26	55.2	R12 000.00	Active Events	6
3	Cape Town Challenge	2025/01/18	40.6	R20 000.00	Runners United	1
4	Durban Ultra Marathon	2025/02/19	45.8	R10 000.00	Endurance Sports SA	2

Voorbeeld van die eerste vier rekords in die **tblMarathons**-tabel nadat die organiseerder, Active Events, uitgegee is:

MarathonID	MarathonName	MarathonDate	Distance	PrizeMoney	Organiser	LocationID
1	East London Marathon	2024/09/12	28.4	R50 000.00	Running South Africa	15
3	Cape Town Challenge	2025/01/18	40.6	R20 000.00	Runners United	1
4	Durban Ultra Marathon	2025/02/19	45.8	R10 000.00	Endurance Sports SA	2
5	Johannesburg Run Fest	2025/03/15	75.9	R13 000.00	Outdoor Adventures	6

(5)

2.2.2 Knoppie [2.2.2 - Qualifying marathons]

Die hardlopers wat vir langafstand-marathons oefen, moet aan kwalifiserende marathons deelneem om vir inskrywing in aanmerking te kom. 'n Kwalifiserende marathon is enige marathon wat oor 'n afstand van 40 km of meer afgelê word.

'n Stad se naam moet ingesleutel word deur 'n toevoerdialoogblokkie ('input dialogue box') te gebruik en dan moet al die kwalifiserende marathons in daardie stad vertoon word.

Kode is voorsien om:

- Die inhoud van die **redQ2_2_2**-komponent skoon te maak
- Die naam van die stad wat ingesleutel is, te onttrek en in die veranderlike **sCity** te stoor

Gebruik die **redQ2_2_2**-komponent om die afvoer te vertoon.

Skryf kode om die volgende te doen:

- Vertoon die naam van die marathon en die afstand van al die kwalifiserende marathons uit die **tblMarathons**-tabel vir die naam van die stad wat ingelees is, deur die gegewe toevoerdialoogblokkie te gebruik.
- Vertoon 'n geskikte boodskap as die naam van die stad wat ingesleutel is, nie in die **tblLocations**-tabel opgespoor kon word nie.

Voorbeeld van afvoer indien Paarl as toevoer ingesleutel is:

2.2.2	
2.2.2 - Qualifying marathons	
Paarl Peak Pursuit	40.2
Worcester Wild Run	48.1

Voorbeeld van afvoer indien Sasolburg as toevoer ingesleutel is:

2.2.2	
2.2.2 - Qualifying marathons	
Sasolburg was not found.	

(13)

- Sleutel jou eksamennummer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING B: 40

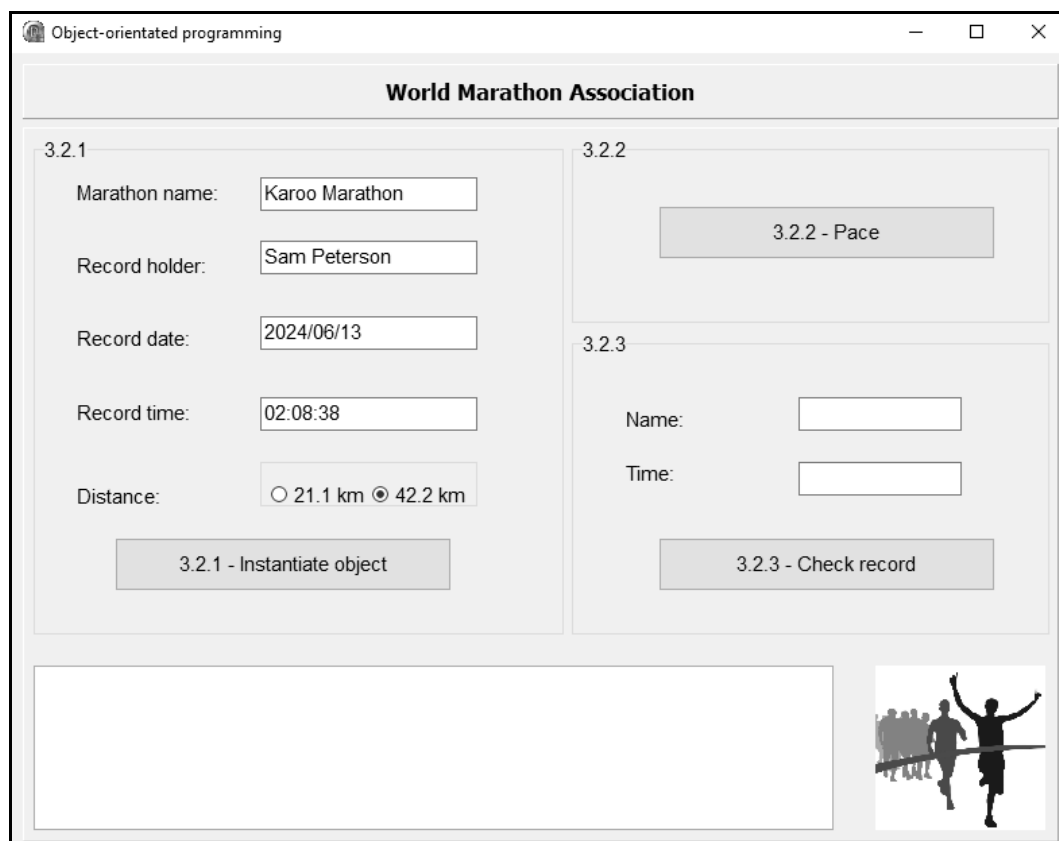
AFDELING C**VRAAG 3: OBJEK-GEORIËNTEERDE PROGRAMMERING**

Die Wêreld Marathon Assosiasie het jou gevra om hulle met die bestuur van die inligting oor rekordhouers van marathons by te staan.

Doen die volgende:

- Maak die onvolledige program in die **Question 3**-lêergids oop.
- Maak die onvolledige objekklas **MRecord_U.pas** oop.
- Sleutel jou eksamennommer as kommentaar in die eerste reël van beide die **Question3_U.pas**-lêer en die **MRecord_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')):



- Voltooi die kode soos gespesifiseer in VRAAG 3.1 en VRAAG 3.2 wat volg.

LET WEL: Jy mag NIE enige addisionele attribute of gebruikergedefinieerde metodes byvoeg NIE, tensy dit uitdruklik in die vraag genoem word.

- 3.1 Die onvoltooide objekklas (**MRecord**) wat voorsien is, bevat die verklaring van vyf attribute wat 'n **Marathon Record**-objek beskryf.

Die attribute van 'n **Marathon Record**-objek is soos volg verklaar:

Attribuut	Tipe	Beskrywing
fMarathonName	String	Die naam van die marathon
fRecordHolder	String	Die naam van die persoon wat tans die rekordhouer is
fRecordDate	String	Die datum van die rekord in die formaat jjjj/MM/dd
fRecordTime	String	Die rekordtyd in die formaat hh:mm:ss
fDistance	Real	Die afstand van die marathon

Die volgende metodes is ook voorsien:

- 'n Gedeeltelik voltooide **konstruktor**-metode
- Die wysigingsmetodes ('mutator methods') **setRecordHolder**, **setRecordDate** en **setRecordTime**
- 'n Hulpmetode ('auxiliary method') **toMinutes** wat die tyd van 'n hardloper as 'n stringparameter in die formaat hh:mm:ss ontvang en die rekordtyd in minute as 'n reële datatipe terugstuur

Voltooi die kode in die objekklas soos wat in VRAAG 3.1.1 tot VRAAG 3.1.5 hieronder beskryf word.

- 3.1.1 Die onvoltooide **konstruktor**-metode ontvang vyf parameters.

Skryf kode om die parameters aan die **fMarathonName**-, **fRecordHolder**-, **fRecordDate**-, **fRecordTime**- en **fDistance**-attribute onderskeidelik toe te ken. (3)

- 3.1.2 Skryf kode vir 'n toegangsmetode ('accessor method') met die naam **getRecordTime** om die **fRecordTime**-attribuut terug te stuur. (2)

- 3.1.3 Skryf kode vir 'n metode met die naam **checkRecord** wat 'n hardloper se tyd as 'n stringparameter ontvang en 'n Boolese waarde terugstuur.

Die metode moet die huidige rekordtyd wat in die **fRecordTime**-attribuut gestoor is, vergelyk met die tyd wat as 'n parameter ontvang is. Stuur 'True' terug indien die tyd van die hardloper wat as 'n parameter ontvang is, minder is as die waarde in die **fRecordTime**-attribuut en 'False' indien nie. (5)

- 3.1.4 Die hardloper se pas ('pace') is die spoed waarteen 'n hardloper 'n marathon voltooi, gemeet in minute per kilometer.

Die volgende formule kan gebruik word om 'n hardloper se pas te bereken:

$$pace = \frac{minutes}{distance}$$

Skryf kode vir 'n metode met die naam **calcPace** wat die gegewe **toMinutes**-metode en die gegewe formule gebruik om die hardloper se pas te bereken en as 'n reële waarde terug te stuur.

(4)

- 3.1.5 Skryf kode vir 'n metode met die naam **toString** om 'n string in die volgende formaat terug te stuur:

```
<fMarathonName> - <fDistance> km: <fRecordHolder>
(<fRecordTime> on <fRecordDate>)
```

Voorbeeld:

```
Karoo Marathon - 42.2 km: Sam Peterson (02:08:38 on
2024/06/13)
```

(5)

- 3.2 'n Onvolledige program is in die **Question 3**-lêergids voorsien. Die program bevat kode vir die objekklas om toeganklik te wees en verklaar 'n objek-veranderlike met die naam **objMRecord**.

Kode om die afvoerarea skoon te maak, is in **btnQ3_2_1**, **btnQ3_2_2** en **btnQ3_3_3** voorsien.

Skryf kode om die take wat beskryf word in VRAAG 3.2.1 tot VRAAG 3.2.3 wat volg, uit te voer.

3.2.1 Knoppie [3.2.1 - Instantiate object]

Sam Peterson is die huidige rekordhouer van die Karoo Marathon. Die huidige rekordinligting vir die Karoo Marathon is soos volg aangeteken:

Marathonnaam	Karoo Marathon
Rekordhouer	Sam Peterson
Datum waarop rekord opgestel is	2024/06/13
Marathon-rekordtyd	02:08:38
Afstand van marathon	42.2 km

Kode is voorsien om die naam van die marathon, die rekordhouer se naam, die datum waarop die rekord opgestel is en die tyd wat vir die rekord aangeteken is, uit die relevante redigeerblokkies te onttrek.

Skryf kode om die volgende te doen:

- Onttrek die afstand van die marathon as 'n reële waarde uit die radiogroep ('radio group') **rgpQ3_2_1**. Die eenheid 'km' moet nie as deel van die afstand se waarde gestoor word nie.
- Gebruik die inligting wat onttrek is om 'n **objMRecord**-objek te instansieer.
- Gebruik die **toString**-metode om die besonderhede van die objek in die **redQ3**-afvoerarea te vertoon.

Voorbeeld van toevoer en afvoer:

3.2.1

Marathon name:

Record holder:

Record date:

Record time:

Distance: 21.1 km 42.2 km

Karoo Marathon - 42.2 km: Sam Peterson (02:08:38 on 2024/06/13)

(8)

3.2.2 Knoppie [3.2.2 - Pace]

Die rekordhouer se pas moet bereken word.

Skryf kode om die **calcPace**-metode te roep en die resultaat in die **redQ3**-komponent te vertoon, afgerond tot DRIE desimale plekke, in die volgende formaat:

Record holder's pace: <Pas van die rekordhouer> min/km

Voorbeeld van toevoer en afvoer:

3.2.1

Marathon name:

Record holder:

Record date:

Record time:

Distance: 21.1 km 42.2 km

Record holder's pace: 3.048 min/km

(4)

3.2.3 Knoppie [3.2.3 - Check record]

Die inligting van die huidige rekordhouer-objek moet vervang word indien 'n ander hardloper die marathon in 'n korter tyd voltooi.

Die gebruiker moet die naam en tyd wat dit die nuwe hardloper geneem het om die marathon te voltooi, in die onderskeie redigeerblokkies **edtQ3_2_3_Name** en **edtQ3_2_3_Time** insleutel.

Skryf kode om die volgende te doen:

- Roep die **checkRecord**-metode met die nuwe hardloper se tyd as 'n argument om te bepaal of die tyd wat vir die nuwe hardloper ingesleutel is, minder as die huidige rekordtyd van die objek is.
- Indien die nuwe hardloper se tyd minder as die huidige rekordtyd is:
 - Dateer die huidige rekordobjek op deur die volgende metodes en argumente te gebruik:

Metodes	Argumente
setRecordHolder	Nuwe hardloper se naam
setRecordTime	Nuwe hardloper se tyd
setRecordDate	Stelseldatum

- Vertoon die opgedateerde objek-inligting in die **redQ3**-afvoerarea deur die **toString**-metode te gebruik.

- Indien die tyd van die nuwe hardloper meer is as die huidige tyd van die objek, vertoon die huidige rekordtyd in die **redQ3**-komponent in die volgende formaat:

The current record time remains: <Huidige rekordtyd>

Voorbeeld van toevoer en afvoer indien die nuwe hardloper se tyd **meer** as die huidige rekordtyd is:



3.2.3

Name:

Time:

3.2.3 - Check record

The current record time remains: 02:08:38

Voorbeeld van toevoer en afvoer indien die nuwe hardloper se tyd **minder** as die huidige rekordtyd is:



3.2.3

Name:

Time:

3.2.3 - Check record

Karoo Marathon - 42.2 km: Adam Smith (02:07:15 on 2024/10/05)

(9)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die objekklas en die vormklas in.
- Stoor jou program.
- Druk die kode in die objekklas en die vormklas indien dit vereis word.

TOTAAL AFDELING C: 40

AFDELING D**VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING**

Doen die volgende:

- Maak die onvolledige program in die **Question 4**-lêergids oop.
- Sleutel jou eksamennommer as kommentaar in die eerste reël van die **Question4_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')):



Die volgende skikkings ('arrays') is in die program voorsien:

- 'n Skikking, **arrMarathons**, wat die name van tien marathons wat in die verlede deur 'n deelnemer voltooi is, bevat. Let daarop dat 'n deelnemer dalk meer as een keer aan 'n marathon deelgeneem het.

```
arrMarathons: array [1 .. 10] of String = ('Wally Hayward',  
'Sasol', 'Soweto', 'Jacaranda City', 'Sasol', 'Durban City', 'Soweto',  
'Soweto', 'Wally Hayward', 'Soweto');
```

- 'n Tweedimensionele skikking, **arrChar**, wat alfabetiese karakters stoor. Die name van nege verskillende marathons is tussen die ander karakters in die rye van hierdie skikking versteek.

```
arrChar: array [1 .. 14, 1 .. 14] of char = (
('u', 'x', 'v', 'm', 's', 'a', 's', 'o', 'l', 'f', 'k', 'j', 't', 'r'),
('u', 'm', 'g', 'e', 'n', 'i', 'w', 'a', 't', 'e', 'r', 'd', 's', 'e'),
('g', 'v', 'o', 'e', 't', 'v', 'a', 'n', 'a', 'f', 'r', 'i', 'k', 'a'),
('e', 'p', 'o', 'y', 'i', 'l', 'c', 'k', 'h', 'j', 's', 'd', 'f', 'd'),
('n', 'k', 'n', 'y', 's', 'n', 'a', 'f', 'o', 'r', 'e', 's', 't', 'u'),
('i', 's', 'y', 'd', 'b', 'c', 'r', 'g', 'h', 'k', 'c', 's', 'a', 'r'),
('w', 'a', 'l', 'l', 'y', 'h', 'a', 'y', 'w', 'a', 'r', 'd', 's', 'b'),
('a', 's', 'q', 'r', 't', 'n', 'n', 'j', 'h', 'e', 'r', 't', 'h', 'a'),
('t', 'o', 'e', 'r', 'y', 'b', 'd', 'r', 'h', 'k', 'l', 'g', 'd', 'n'),
('e', 'j', 'a', 'c', 'a', 'r', 'a', 'n', 'd', 'a', 'c', 'i', 't', 'y'),
('r', 'y', 'j', 'f', 'g', 'f', 'c', 'f', 'g', 'u', 'h', 'v', 'c', 'i'),
('k', 'h', 'h', 'l', 'p', 'h', 'i', 'l', 'l', 'c', 'r', 'e', 's', 't'),
('a', 'd', 'e', 'v', 'd', 's', 'o', 'w', 'e', 't', 'o', 'm', 'k', 'y'),
('p', 'd', 'u', 'r', 'b', 'a', 'n', 'c', 'i', 't', 'y', 'z', 'c',
'l'));
```

Voltooi die kode vir elke afdeling van VRAAG 4, soos in VRAAG 4.1 en VRAAG 4.2 hieronder beskryf word.

4.1 Knoppie [4.1 - Count marathons]

Gebruik die gegewe skikking met die naam **arrMarathons** wat die name van tien marathons, waaraan 'n hardloper die afgelope vyf jaar deelgeneem het, bevat. Daar is 'n herhaling van marathonname in die skikking aangesien die hardloper meer as een keer aan sommige van die marathons deelgeneem het.

Kode is voorsien vir die inlynstelling en om die opskrifte in die **redQ4_1** 'rich edit'-komponent te voltooi.

Skryf kode om die volgende te doen:

Vertoon die marathon se naam en die getal kere wat die hardloper aan elke marathon deelgeneem het in die **redQ4_1**-komponent, soos in die afvoer hieronder getoon.

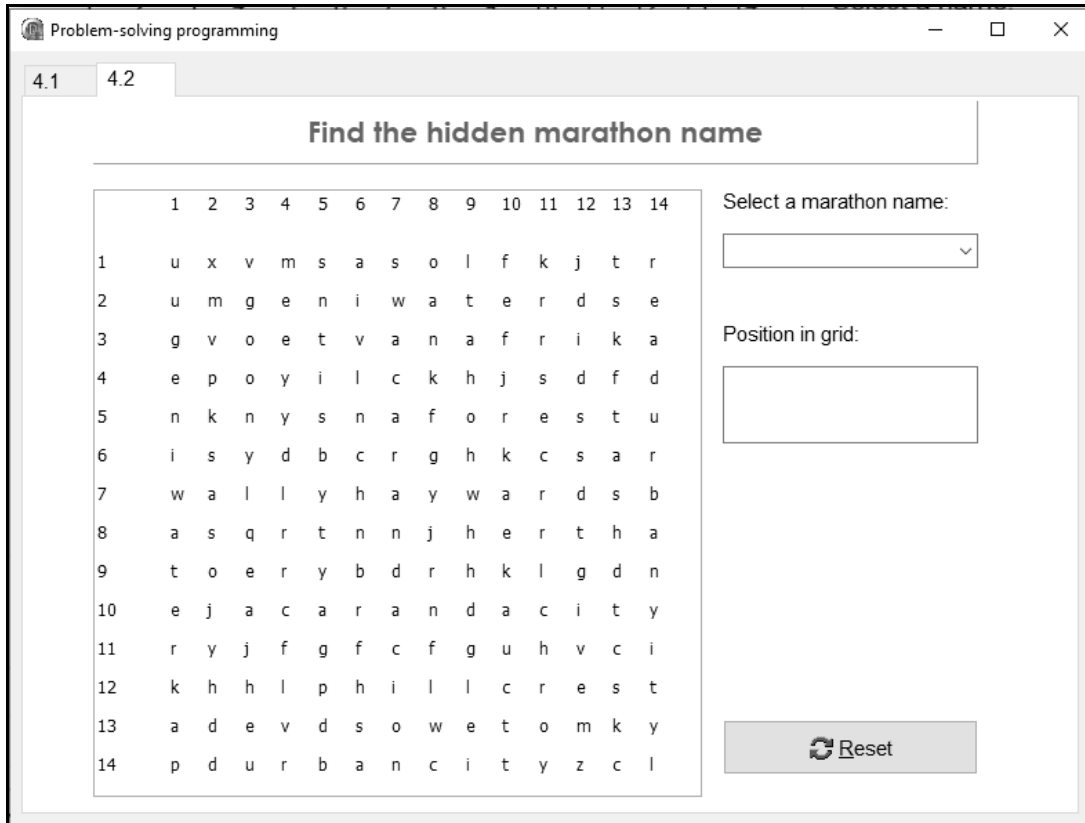
Voorbeeld van afvoer vir die inhoud in die skikking **arrMarathons**:

Marathon name	Number of times
Wally Hayward	2
Sasol	2
Soweto	4
Jacaranda City	1
Durban City	1

(11)

- 4.2 'n Raaiselwoordspeletjie is ontwikkel om bewustheid van die verskillende marathons te skep.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')):



LET WEL: Kode is voorsien vir 'n metode met die naam **display2D** wat die skikking **arrChar** in die 'rich edit' **redQ4_2** vertoon.

Kombinasieblokkie [cmbQ4_2]

Die raaiselwoordspeletjie gebruik karakters wat in die gegewe tweedimensionele skikking met die naam **arrChar** gestoor word en wat 14 rye en 14 kolomme bevat. Nege marathonname is in die kombinasieblokkie **cmbQ4_2** voorsien en hierdie name is ook tussen die karakters in die rye van die skikking **arrChar** weggesteek. Die versteekte marathonname is horisontaal van links na regs in die rye van die skikking **arrChar** geplaas.

Skryf kode om die volgende te doen:

- Onttrek die geselekteerde marathonnaam uit die kombinasieblokkie **cmbQ4_2**.
- Vind die geselekteerde marathonnaam wat in die skikking **arrChar** versteek is.
- Gebruik die **memQ4_2** om die ry- en kolomwaardes van die geselekteerde woord in die volgende formaat te vertoon:

Row <Ry-indeks> @ column <Begin-kolomindeks> to <Einde-kolomindeks>

- Verander die karakters van die geselekteerde marathonnaam in die skikking **arrChar** na hoofletters.
- Roep die **display2D**-metode om die **arrChar**-skikking te vertoon om die geselekteerde woord in **hoofletters** te toon.

Voorbeeld van toevoer en afvoer indien die marathonnaam 'knysnaforest' in die kombinasieblokkie **cmbQ4_2** geselekteer is:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	u	x	v	m	s	a	s	o	l	f	k	j	t	r
2	u	m	g	e	n	i	w	a	t	e	r	d	s	e
3	g	v	o	e	t	v	a	n	a	f	r	i	k	a
4	e	p	o	y	i	l	c	k	h	j	s	d	f	d
5	n	K	N	Y	S	N	A	F	O	R	E	S	T	u
6	i	s	y	d	b	c	r	g	h	k	c	s	a	r

Select a marathon name:

Position in grid:

(19)

LET WEL: 'n **Reset**-knoppie is voorsien om die karakters in die tweedimensionele skikking na kleinletters terug te stel.

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING D: 30
GROOTTOTAAL: 150

INLIGTINGSTEGNOLOGIE V1**DATABASIS-INLIGTING VIR VRAAG 2:**

Die ontwerp van die databasistabelle is soos volg:

Tabel: **tblLocations**

Hierdie tabel bevat die besonderhede van elke plek.

Veldnaam	Datatype	Beskrywing
LocationID (PK)	Number	'n Unieke ID wat 'n plek identifiseer
City	Text (20)	Die naam van die stad waar die plek is
Province	Text (15)	Die naam van die provinsie waar die stad geleë is
Altitude	Number	Die hoogte bo seespieël van die plek
Population	Number	Die bevolking van die stad

Voorbeeld van die eerste tien rekords van die **tblLocations**-tabel:

LocationID	City	Province	Altitude	Population
1	Cape Town	Western Cape	42	4618000
2	Durban	KwaZulu-Natal	8	3721000
3	Bloemfontein	Free State	1395	598000
4	Pretoria	Gauteng	1339	2473000
5	Gqeberha	Eastern Cape	85	967677
6	Johannesburg	Gauteng	1753	5635000
7	Rustenburg	North West	1170	574000
8	Polokwane	Limpopo	1310	479000
9	Mbombela	Mpumalanga	1653	110159
10	eMalahleni	Mpumalanga	1520	434238

Tabel: **tblMarathons**

Hierdie tabel bevat inligting oor beskikbare marathons in elke plek.

Veldnaam	Datatype	Beskrywing
MarathonID (PK)	Number	'n Unieke ID wat die marathon identifiseer
MarathonName	Text (30)	Die naam van die marathon
MarathonDate	Date/Time	Die datum waarop die marathon plaasvind
Distance	Number	Die afstand van elke marathon
PrizeMoney	Currency	Die prysgeld wat aan die wenner van die marathon toegeken word
Organiser	Text (25)	Die naam van die organiseerder van die marathon
LocationID (FK)	Number	Die ID van die plek waar die marathon plaasvind

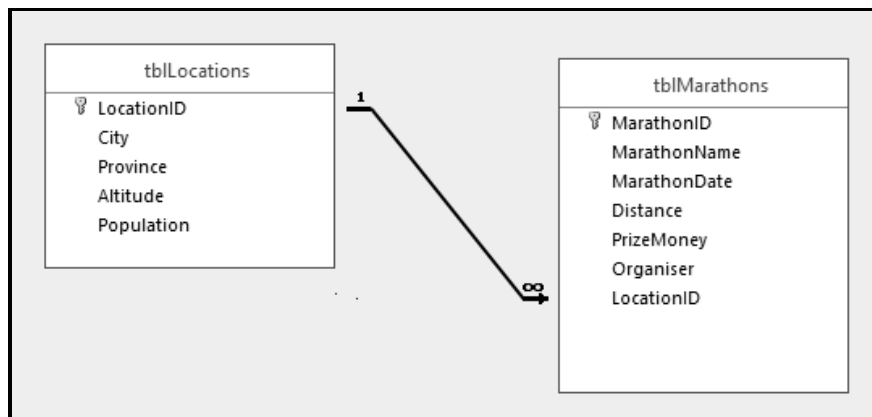
Voorbeeld van die eerste tien rekords van die **tblMarathons**-tabel:

MarathonID	MarathonName	MarathonDate	Distance	PrizeMoney	Organiser	LocationID
1	East London Marathon	2024/09/12	28.4	R50 000.00	Running South Africa	15
2	Joburg Run	2024/10/26	55.2	R12 000.00	Active Events	6
3	Cape Town Challenge	2025/01/18	40.6	R20 000.00	Runners United	1
4	Durban Ultra Marathon	2025/02/19	45.8	R10 000.00	Endurance Sports SA	2
5	Johannesburg Run Fest	2025/03/15	75.9	R13 000.00	Outdoor Adventures	6
6	Pretoria Classic	2025/04/28	32.5	R8 000.00	PE Running Club	4
7	Port Elizabeth Pursuit	2025/05/10	90.5	R5 000.00	Wild Runners SA	5
8	Kruger National Challenge	2025/06/23	58.3	R30 000.00	Mountain Marathoners	9
9	Drakensberg Dash	2025/07/07	37.6	R35 000.00	Winelands Athletics	14
10	Soweto Sprint	2025/08/19	42.1	R2 500.00	City Runners	6

LET WEL:

- Konneksiekode is voorsien.
- Die databasis word met 'n wagwoord beskerm, dus sal jy nie direkte toegang tot die databasis kan verkry nie.

Die volgende een-tot-baie-verwantskap met referensiële integriteit ('referential integrity') bestaan tussen die twee tabelle in die databasis:



BEPLANNINGSBLADSY 1

BEPLANNINGSBLADSY 2

Eksamenplakker

150

INLIGTINGSTEGNOLOGIE V1 – NOVEMBER 2024
INLIGTINGSBLAD (*moet NÁ die 3 uur-sessie deur die kandidaat ingevul word*)

SENTRUMNOMMER _____

EKSAMENNOMMER _____

WERKSTASIENOMMER _____

Weergawe van Delphi wat tydens die INLIGTINGSTEGNOLOGIE NSS NOV 2024-eksamen gebruik is:

Maak 'n kruisie in die toepaslike blokkie (X)	Delphi 10	Delphi XE	Delphi 10.3	Delphi Community	Delphi 11	Delphi 12	Ander: (Spesifiseer) _____
---	-----------	-----------	-------------	------------------	-----------	-----------	----------------------------

NAAM VAN LÊERGIDS ('FOLDER') _____

Kandidaat moet die lêernaam/-name vir elke antwoord invul. Maak 'n regmerkie indien gestoor en/of gepoog.

Vraag-nommer	Lêernaam	Gestoor (✓)	Gepoog (✓)	Maksimum Punt	Punt Verdien	Nasiener Voorletters/ Kode
1	Question1_P.dproj			40		
2	Question2_P.dproj			40		
3	MRecord_U.pas			19		
	Question3_P.dproj			21		
4	Question4_P.dproj			30		
TOTAAL				150		

Kommentaar: (*slegs vir kantoor/nasiener se gebruik*)



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEKNOLOGIE V1

NOVEMBER 2024

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 33 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as die basis vir die nasiensessie gebruik word. Dit is voorberei om deur nasieners gebruik te word. Daar word van alle nasieners verwag om 'n deeglike standaardiseringsvergadering by te woon om seker te maak dat die riglyne konsekwent geïnterpreteer en tydens die nasien van die leerders se werk toegepas word.
- Let op dat leerders wat 'n alternatiewe korrekte oplossing as wat as voorbeeld van 'n oplossing in die nasienriglyne gegee word verskaf, volle krediet vir die relevante oplossing moet kry tensy die spesifieke instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom is nie.
- **Bylae A, B, C en D** (bladsy 3 tot 15) sluit die nasienriglyne vir elke vraag in.
- **Bylae E, F, G en H** (bladsy 16 tot 33) bevat voorbeelde van oplossings vir Vrae 1 tot 4 in programmeringskode.
- Kopieë van **Bylae A, B, C, D en die opsomming van die leerder se punte** (bladsy 3 tot 15) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

BYLAE A**VRAAG 1: NASIENRUBRIEK – ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1.1	<p>Knoppie [1.1 - Random]</p> <p>Genereer ewekansige getal ✓ van 5 tot 10 (beide waardes ingesluit in reeks) ✓ edtQ1_1_1.Text := ✓ skakel getal om na String ✓</p>	4	
1.1.2	<p>Knoppie [1.1.2 - Round up]</p> <p>spnQ1_1_2.Value := ✓ Opgerond ✓ Ceil (NUMBER) Round (NUMBER + 0.5) Floor (NUMBER) + 1 Trunc (NUMBER) + 1 Met konstante as parameter ✓</p>	3	
1.2	<p>Knoppie [1.2 - Surface area]</p> <p>Onttrek die hoogte en radius ✓ uit die redigeerblokkies Omgeskakel na Float ✓ <i>Bereken die oppervlakte</i> $A := PI * rR * (rR + \sqrt{\text{sqr}(rR) + \text{sqr}(rH)})$ ✓ Vertoon die oppervlakte in die etiket ✓ geformateer tot 2 desimale plekke ✓ FormatFloat('0.00', A); FloatToStrF(A, ffFixed, 10, 2); Format('%f', [A]);</p> <p>AANVAAR ook Power(rR, 2) in plaas van Sqr</p> <p>LET WEL: Hakkies moet korrek ingevoeg wees in die berekening.</p>	8	

1.3	<p>Knoppie [1.3 - Read file]</p> <p>Verklaar veranderlikes vir die adres en slaapkamers ✓ AssignFile(tFile, 'Houses.txt') ✓ Reset(tFile) ✓</p> <p>Lus deur die tekslêer met die regte voorwarde ✓ Readln ✓ (tFile, adres-veranderlike ✓) Readln(tFile, slaapkamer-veranderlike) ✓ Voeg die adres en slaapkamer saam met 'n strepie (-) tussen die adres en slaapkamer ✓ Vertoon afvoer in die 'rich edit' ✓</p> <p>AANVAAR ook alternatiewe om uit lêer te lees: As onewe reëlnommer, stoor waarde in adres-veranderlike (2) As ewe reëlnommer, stoor waarde in slaapkamer-veranderlike (1)</p>	9	
1.4	<p>Knoppie [1.4 - Add name]</p> <p>Ottrek die naam uit die kombinasielys ✓</p> <p>Toets of ✓ die kontroleblokkie gemerk is ✓ Voeg naam by die reeds-betaal 'rich edit'-komponent ✓ Anders ✓ Voeg naam by die nie-betaal 'rich edit'-komponent ✓</p> <p>Verwyder die geselekteerde naam van die kombinasielys ✓ cmbQ1_4.DeleteSelected cmbQ1_4.Items.Delete(cmbQ1_4.ItemIndex)</p> <p>LET WEL: GEEN punt indien die item in die kombinasielys vervang word met 'n leë string en nie verwyder word nie.</p>	7	

1.5	<p>Knoppie [1.5 - Replace]</p> <p>Lus deur die string ✓ Toets of karakter NIE 'n spasie is NIE ✓ Voeg karakter by wagwoord-veranderlike in omgekeerde volgorde ✓ Vertoon wagwoord in memo ✓</p> <p>Lus Indeks vanaf 1 tot lengte van wagwoord ✓ Indien Indeks MOD 3 = 0 ✓ Genereer ewekansige getal in reeks 1 tot 6 ✓ Vervang wagwoord-karakter by Indeks met ewekansige karakter uit sCharacters ✓</p> <p>Vertoon opgedateerde wagwoord in memo ✓</p> <p>Alternatief vir die eerste 3 punte: Gebruik lus om eers spasies te verwyder (1), dan lus (1) deur die veranderde string in omgekeerde volgorde (1)</p> <p>In die wagwoord gedeelte AANVAAR ook:</p> <ul style="list-style-type: none"> • Lus-indeks vanaf 3 tot lengte van wagwoord • Genereer ewekansige waarde van 1 tot Length(sCharacter) 	9	
	TOTAAL AFDELING A:	40	

BYLAE B

VRAAG 2: NASIENRUBRIEK – DATABASISPROGRAMMERING

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	SQL-stellings		
2.1.1	Knoppie [2.1.1 - Low population] SELECT * ✓ FROM tblLocations ✓ WHERE Population < 200000 ✓	3	
2.1.2	Knoppie [2.1.2 - Runners United September runs] SELECT MarathonID, MarathonDate, Distance FROM tblMarathons ✓ WHERE Organiser = "Runners United" ✓ AND ✓ Month(MarathonDate) = 9 ✓ AANVAAR ook: <ul style="list-style-type: none"> • Organiser LIKE "%Runners United%" • MarathonDate LIKE "%/09/%" • MID(MarathonDate, 6, 2) = 9 	4	
2.1.3	Knoppie [2.1.3 - Marathon locations] SELECT City & " - " ✓ & LEFT(Province, 3) ✓ AS [Location] ✓ FROM tblLocations ✓ AANVAAR ook: <ul style="list-style-type: none"> • + in die plek van & • Mid(Province, 1, 3) 	4	
2.1.4	Knoppie [2.1.4 - Add city] INSERT INTO tblLocations ✓ VALUES ✓ (19, "Welkom", "Free State", 1198, 423016) ✓✓ (korrekte volgorde (1 punt), korrekte getal parameters (1 punt))	4	

2.1.5	Knoppie [2.1.5 - City details] <pre> SELECT City, ✓ COUNT(City) ✓ AS NumMarathons, SUM(Prizemoney) AS [Total Prize Money] ✓ FROM tblMarathons , tblLocations ✓ WHERE tblMarathons.LocationID = tblLocations.LocationID ✓ GROUP BY City ✓ HAVING SUM(Prizemoney) > 50000 ✓ </pre> AANVAAR ook: <ul style="list-style-type: none"> • Count(*) • Count('n Veldnaam) / Count(tblLocations.LocationID) 	7	
	Subtotaal:	22	

VRAAG 2: NASIENRUBRIEK (VERVOLG)

2.2	Databasismanipulasie		
2.2.1	Knoppie [2.2.1 - Remove marathons]		
	<p>Gaan na die eerste rekord in tblMarathons ✓ Stap met lus ('loop') deur tblMarathons ✓ Toets of tblMarathons ['Organiser'] = sOrganiser ✓ tblMarathons.Delete ✓ anders tblMarathons.Next ✓ End lus</p>	5	
2.2.2	Knoppie [2.2.2 - Qualifying events]		
	<p>Inisialiseer vlag / teller ✓ tblLocations.First (merk saam met tblLocations.next) ** ✓ Stap met lus ('loop') deur tblLocations ✓ Toets of tblLocations['City'] = sCity ✓ Verander vlag / inkrementeer teller ✓ tblMarathons.First (merk met tblMarathons.next) ** ✓ Stap met lus ('loop') deur tblMarathons ✓ Toets of (tblMarathons['LocationID'] = tblLocations['LocationID']) ✓ AND ✓ (tblMarathons['Distance'] >= 40) ✓ Vertoon die MarathonName en Distance omgeskakel na 'n string ✓ tblMarathons.Next Einde lus (tblMarathons) tblLocations.Next Einde lus (tblLocations)</p> <p>Toets vlag / teller ✓ Vertoon boodskap wat aandui dat die stad nie gevind is nie ✓</p> <p>LET WEL:**</p> <ul style="list-style-type: none"> • Die FIRST- en NEXT-stellings van die buitenste lus (vir tblLocations) moet beide in die korrekte posisie wees vir een punt • Die FIRST- en NEXT-stellings van die binneste lus (vir tblMarathons) moet beide in die korrekte posisie wees vir een punt 	13	
	Subtotaal:	18	
	TOTAAL AFDELING B:	40	

BYLAE C**VRAAG 3: NASIENRUBRIEK – OBJEK-GEÖRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	<p>Konstruktor Create</p> <p>Ken die korrekte parameters toe aan die attribute fMarathonName en fRecordHolder ✓ Ken die korrekte parameters toe aan fRecordDate en fRecordTime ✓ Ken die korrekte parameter toe aan fDistance ✓</p>	3	
3.1.2	<p>Funksie getRecordTime</p> <p>Opskrif van funksie met string as terugstuur datatipe ✓ Stuur fRecordTime terug ✓</p>	2	
3.1.3	<p>Funksie checkRecord</p> <p>Opskrif van funksie met Boolean terugstuur datatipe ✓ met string parameter ✓ Toets of parameterwaarde < fRecordTime ✓ Stuur true terug ✓ Anders Stuur false terug ✓</p> <p>AANVAAR ook:</p> <ul style="list-style-type: none"> • StrToTime and toMinutes to compare the times. • Toets of fRecordTime > parameterwaarde 	5	
3.1.4	<p>Funksie calcPace</p> <p>Opskrif van funksie met reële terugstuur tipe ✓ Stuur terug ✓ toMinute(fRecordTime) ✓ / fDistance ✓</p>	4	
3.1.5	<p>Funksie toString met string terugstuur tipe</p> <p>Opskrif van funksie met string as terugstuur tipe ✓ Bou string met streep, 'km:', 'on' en hakkies ✓ Afstand omgeskakel na string ✓ Bevat al die attribute ✓ Stuur die string terug ✓</p>	5	
	Subtotaal: Objekklas	19	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	<p>Knoppie [3.2.1 - Instantiate object]</p> <p>Afstand onttrek uit radiogroep: Items by ItemIndex ✓</p> <p>Onttrek die afstand deur gebruik te maak van String-manipulering ✓ omgeskakel na reël ✓</p> <p>objMRecord:= ✓ TMRecord✓ .create ✓ (marathon name, record holder, record date, record time, distance) ✓</p> <p>Vertoon objek in redQ3 deur gebruik van toString-metode ✓</p> <p>Alternatief om die afstand te onttrek: Gebruik if else / case (1) en ken afstand toe (1)</p>	8	
3.2.2	<p>Knoppie [3.2.2 - Pace]</p> <p>Roep calcPace-metode ✓</p> <p>Vertoon die resultaat in redQ3 omgeskakel na 'n string ✓</p> <p>Geformatteeer tot 3 desimale plekke ✓ met korrekte byskrif (min/km) ✓</p>	4	
3.2.3	<p>Knoppie [3.2.3 - Check record]</p> <p>Verkry naam en tyd van atleet uit redigeerblokkies ✓</p> <p>As objMRecord.checkRecord ✓ (Tyd van atleet) ✓</p> <p>Roep setRecordHolder (Name van atleet) ✓</p> <p>Roep setRecordTime (Tyd van atleet) ✓</p> <p>Roep setRecordDate ✓ (Huidige datum as string) ✓</p> <p>Vertoon met behulp van toString-metode ✓</p> <p>Anders Vertoon huidige rekordtyd in redQ3 deur die getRecordTime-metode te roep ✓</p>	9	
	Subtotaal Vormklas:	21	
	TOTAAL AFDELING C:	40	

	<p>Konsep 2: (Skep lus (array/string) sonder duplikate)</p> <p>Bou string / vul skikking met unieke waardes [4] Inisialiseer teller en bFound (1) Buite en binne lusse (1) IF-stellings wat in toetse gebruik word (1) Toekenningstellings (1)</p> <p>Lus deur unieke waardes in Temp skikking en tel in arrMarathons skikking [5]</p> <p>Lus x deur tydelike skikking met unieke waardes / string (1) Inisialiseer iNumMarathons (1) Lus y deur arrMarathons (1) Toets of arrTemp[x] gelyk aan arrMarathons[y] (1) Inkrementeer iNumMarathons (1)</p> <p>Vertoon [2]</p> <p> Vertoon marathon se naam (1) en teller (1)</p> <p>Konsep 3: (Tydelike skikkings met unieke marathon name en teller waardes)</p> <p>Gebruik arrTempMarathons en arrCountMarathons</p> <p>Inisialiseer teller ** (1) met found := false Buite lus deur skikking (1) Stel bFound op FALSE ** Binne lus van 1 to teller (1) Toets arrMarathons[buite] = arrTempMarathons[binne] (1) Verander bFound na TRUE (1) Vermeerder arrCountMarathons[binne] (1) IF stelling bFound is FALSE (1) Vermeerder teller (1) Stel arrCountMarathons[teller] na 1 (1) ## Stel arrTempMarathons[teller] to rrMarathons[buite] ##</p> <p>Lus van 1 tot teller (1) Vertoon arrTempMarathons en arrCountMarathons (omgeskakel na heelgetal) (1)</p>		
--	---	--	--

4.2	<p>Combobox [cmbQ4_2]</p> <p><i>Onttrek woord uit kombinasie lys [1]</i> Lees woord uit kombinasie lys ✓</p> <p><i>Toets of woord in die ry voorkom [10]</i> Lus deur rye 1 tot 14 ✓ Inisialiseer nuweString ✓ Lus kolom 1 tot 14 ✓ Bou nuweString ✓ met karakters uit array[R,C] ✓ As nuweString = geselekteerde woord ✓ Bereken/Stel beginkolom-indeks ✓ Bereken eindkolom-indeks: Beginkolom-indeks ✓ + lengte van woord ✓ - 1 ✓</p> <p><i>Vertoon die ry-nommer, begin- en eindkolom-indeks [3]</i> Vertoon ry-nommer ✓ beginkolom ✓ en eindkolom ✓</p> <p><i>Verander na hoofletters [4]</i> Lus deur woord in 2D skikking beginkolom-indeks ✓ Eindkolom-indeks ✓ Verander karakter in 2D ✓ na hoofletter-karakter ✓</p> <p><i>Vertoon 2D skikking [1]</i> Roep vertoon metode ✓</p>	19	
-----	--	----	--

	<p>Konsep 1: (Ry- en Kolomlusse – kopieër uit ry)</p> <p><i>Onttrek woord uit kombinasie lys [1]</i> Lees woord uit kombinasie lys (1)</p> <p><i>Toets of woord in die ry voorkom [10]</i> Lus iR van 1 tot 14 (1) Lus iC van 1 tot 14 (1) Toets of marathonnaam in ry iR is deur by kolom-indeks iC te begin (4) Bereken/Stel beginkolom-indeks van woord (1) Bereken eindkolom-indeks van woord (3)</p> <p><i>Vertoon die ry-nommer, begin- en eindkolom-indeks [3]</i> Vertoon ry-nommer (1) beginkolom (1) en eindkolom (1)</p> <p><i>Verander na hoofletters [4]</i> Lus van beginkolom-indeks (1) tot eindkolom (1) Verander karakter (1) na hoofletter-karakter (1)</p> <p><i>Vertoon 2D skikking [1]</i> Roep vertoon metode (1)</p> <p>Konsep 2: (Ry – gebruik pos direk)</p> <p><i>Onttrek woord uit kombinasie lys [1]</i> Lees woord uit kombinasie lys (1)</p> <p><i>Toets of woord in die ry voorkom [10]</i> Ry lus van 1 tot 14 (1) Toets of marathonnaam in die ry is (4) Bereken beginkolom-indeks van woord (2) Bereken eindkolom-indeks van woord (3)</p> <p><i>Vertoon die ry-nommer, begin- en eindkolom-indeks [3]</i> Vertoon ry-nommer (1) beginkolom (1) en eindkolom (1)</p> <p><i>Verander na hoofletters [4]</i> Lus van beginkolom (1) tot eindkolom (1) Verander karakter (1) na hoofletter-karakter (1)</p> <p><i>Vertoon 2D skikking [1]</i> Roep vertoon metode (1)</p>		
	TOTAAL AFDELING D:	30	

OPSOMMING VAN LEERDER SE PUNTE:

SENTRUMNOMMER:		LEERDER SE EKSAMENNUMMER:			
	AFDELING A	AFDELING B	AFDELING C	AFDELING D	
	VRAAG 1	VRAAG 2	VRAAG 3	VRAAG 4	GROOT-TOTAAL
MAKS. PUNTE	40	40	40	30	150
LEERDER SE PUNTE					

BYLAE E: OPLOSSING VIR VRAAG 1

```
unit Question1_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ComCtrls, pngimage, ExtCtrls, Spin, math,
  jpeg;

type
  TfrmQuestion1 = class(TForm)
    grpQ1_1: TGroupBox;
    grpQ1_2: TGroupBox;
    grpQ1_3: TGroupBox;
    grpQ1_4: TGroupBox;
    grpQ1_5: TGroupBox;
    btnQ1_1_1: TButton;
    btnQ1_1_2: TButton;
    edtQ1_1_1: TEdit;
    spnQ1_1_2: TSpinEdit;
    Label1: TLabel;
    Label2: TLabel;
    edtQ1_2_r: TEdit;
    edtQ1_2_h: TEdit;
    btnQ1_2: TButton;
    btnQ1_3: TButton;
    redQ1_3: TRichEdit;
    Label3: TLabel;
    chbQ1_4: TCheckBox;
    btnQ1_4: TButton;
    Label4: TLabel;
    Label5: TLabel;
    redQ1_4_P: TRichEdit;
    redQ1_4_NP: TRichEdit;
    btnQ1_5: TButton;
    cmbQ1_4: TComboBox;
    edtQ1_5: TEdit;
    lblQ1_2: TLabel;
    Image1: TImage;
    memQ1_5: TMemo;
    procedure btnQ1_1_1Click(Sender: TObject);
    procedure btnQ1_1_2Click(Sender: TObject);
    procedure btnQ1_2Click(Sender: TObject);
    procedure btnQ1_3Click(Sender: TObject);
    procedure btnQ1_4Click(Sender: TObject);
    procedure btnQ1_5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion1: TfrmQuestion1;
implementation

Kopiereg voorbehou
```

```
{$R *.dfm}

// =====
// 1.1.1 Random 4 punte
// =====
procedure TfrmQuestion1.btnQ1_1_1Click(Sender: TObject);
begin
  edtQ1_1_1.Text := intToStr(randomRange(5, 11));
end;

// =====
// 1.1.2 Random 3 punte
// =====

procedure TfrmQuestion1.btnQ1_1_2Click(Sender: TObject);
const
  NUMBER = 5.63247;
begin
  spnQ1_1_2.Value := ceil(NUMBER);

end;

// =====
// 1.2 Surface area 8 punte
// =====
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
  rA, rH, rR: real;

begin
  rH := StrToFloat(edtQ1_2_h.Text);
  rR := StrToFloat(edtQ1_2_r.Text);
  rA := PI * rR * (rR + Sqrt(Sqr(rH) + Sqr(rR)));
  lblQ1_2.Caption := FloatToStrF(rA, ffFixed, 8, 2);
end;

// =====
// 1.3 Read file 9 punte
// =====
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
  tFile: textfile;
  sAdd, sRooms: String;
begin
  AssignFile(tFile, 'Houses.txt');
  Reset(tFile);

  while NOT Eof(tFile) do
  begin
    readln(tFile, sAdd);
    readln(tFile, sRooms);
    redQ1_3.Lines.Add(sAdd + ' - ' + sRooms);
  end;
  CloseFile(tFile);
end;
```

```
// =====  
// 1.4 Add name 7 punte  
// =====  
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);  
var  
    sName: String;  
begin  
    sName := cmbQ1_4.Text;  
    if chbQ1_4.Checked then  
    begin  
        redQ1_4_P.Lines.Add(sName);  
    end  
    else  
    begin  
        redQ1_4_NP.Lines.Add(sName);  
    end;  
  
    cmbQ1_4.items.Delete(cmbQ1_4.ItemIndex);  
end;  
// =====  
// 1.5 Replace 9 punte  
// =====  
procedure TfrmQuestion1.btnQ1_5Click(Sender: TObject);  
var  
    sNameSurname, sCharacters, sPassword: String;  
    iCnt, iLen, iRandom: integer;  
begin  
    // Provided code  
    sNameSurname := edtQ1_5.Text;  
    sCharacters := '@#$%^&';  
    // Add your code here  
    sPassword := '';  
    iLen := Length(sNameSurname);  
    for iCnt := iLen downto 1 do  
    begin  
        if sNameSurname[iCnt] <> ' ' then  
            sPassword := sPassword + sNameSurname[iCnt];  
    end;  
    memQ1_5.Lines.Add(sPassword);  
    for iCnt := 1 to length(sPassword) do  
    begin  
        if(iCnt mod 3 = 0) then  
        begin  
            iRandom := random(6) + 1;  
            sPassword[iCnt] := sCharacters[iRandom];  
        end;  
    end;  
    memQ1_5.Lines.Add(sPassword);  
end;  
end.
```

BYLAE F: OPLOSSING VIR VRAAG 2

```
// =====  
// 2.1 - Section: SQL statements  
// =====  
  
// =====  
// 2.1.1 Low population 3 punte  
// =====  
  
procedure TfrmQuestion2.btnQ2_1_1Click(Sender: TObject);  
var  
    sSQL1: String;  
begin  
    // Question 2.1.1  
  
    sSQL1 := 'SELECT * ' + 'FROM tblLocations ' + 'WHERE Population <  
200000';  
  
    // Provided code - do not change  
    dbCONN.runSQL(sSQL1);  
end;  
  
// =====  
// 2.1.2 Runners United September runs 4 punte  
// =====  
  
procedure TfrmQuestion2.btnQ2_1_2Click(Sender: TObject);  
var  
    sSQL2: String;  
begin  
    // Question 2.1.2  
    sSQL2 := 'SELECT MarathonID, MarathonDate, Distance ' +  
        'FROM tblMarathons ' + 'WHERE Organiser = "Runners United" AND ' +  
        'Month(MarathonDate) = 9';  
  
    // Provided code - do not change  
    dbCONN.runSQL(sSQL2);  
end;  
  
// =====  
// 2.1.3 Marathon locations 4 punte  
// =====  
  
procedure TfrmQuestion2.btnQ2_1_3Click(Sender: TObject);  
var  
    sSQL3: String;  
begin  
    // Question 2.1.3  
  
    sSQL3 := 'SELECT City & " - " & left(Province,3) AS [Location] ' +  
        'FROM tblLocations ';  
  
    // Provided code - do not change  
    dbCONN.runSQL(sSQL3);  
end;
```

```
// =====  
// 2.1.4 Add city 4 punte  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_4Click(Sender: TObject);  
var  
    sSQL4: String;  
    bValid: boolean;  
begin  
    // Question 2.1.4  
  
    sSql4 := 'INSERT INTO tblLocations VALUES  
            (19, "Welkom", "Free State", 1198, 423016)';  
  
    // Provided code - do not change  
    dbCONN.ExecuteSQL(sSQL4);  
  
end;
```

```
// =====  
// 2.1.5 High prize money 7 punte  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_5Click(Sender: TObject);  
var  
    sSQL5: String;  
    bChanged: boolean;  
begin  
    // Question 2.1.5  
  
    sSQL5 :=  
        'SELECT City, COUNT(City) AS [NumMarathons],  
        SUM(Prizemoney) AS [Total Prize Money] '  
        + 'FROM tblMarathons , tblLocations ' +  
        'WHERE tblMarathons.LocationID = tblLocations.LocationID ' +  
        'GROUP BY City HAVING SUM(Prizemoney) > 50000';  
  
    // Provided code - do not change  
    dbCONN.runSQL(sSQL5);  
end;
```

```
// =====  
// 2.2 - Section: Delphi code  
// =====  
  
// =====  
// 2.2.1 Remove marathons 5 punte  
// =====  
  
procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);  
var  
    sOrganiser: String;  
begin  
    // Provided code  
    sOrganiser := InputBox('Organiser',  
        'Enter the name of the organiser to remove', 'Endurance Sports SA');  
  
    // Question 2.2.1  
    tblMarathons.First;  
    while NOT tblMarathons.Eof do  
    begin  
        if tblMarathons['Organiser'] = sOrganiser then  
            tblMarathons.Delete  
        else  
            tblMarathons.Next;  
        end;  
    end;  
end;  
  
// =====  
// 2.2.2 Qualifying events 13 punte  
// =====  
  
procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);  
var  
    sCity: String;  
    bFound: boolean;  
    iLocation: Integer;  
begin  
    // Provided code  
    sCity := InputBox('City', 'Enter the name of the city', 'Paarl');  
  
    // Question 2.2.2  
    bFound := False;  
    tblLocations.First;  
    while (NOT tblLocations.Eof) AND (bFound = False) do  
    begin  
        if tblLocations['City'] = sCity then  
        begin  
            bFound := True;  
            iLocation := tblLocations['LocationID'];  
        end;  
        tblLocations.Next;  
    end;  
  
    if bFound then  
    begin  
        tblMarathons.First;
```

```

while NOT tblMarathons.Eof do
begin
  if (tblMarathons['LocationID'] = iLocation) AND
    (tblMarathons['Distance'] >= 40) then
  begin
    redQ2_2_2.Lines.Add(tblMarathons['MarathonName'] + #9 +
      FloatToStr(tblMarathons['Distance']));
  end;
  tblMarathons.Next;
end
end
else
  redQ2_2_2.Lines.Add(sCity + ' is not found.');
```

// Alternative:

```

{ bFound := False;
tblLocations.First;
while NOT tblLocations.Eof do
begin
  if tblLocations['City'] = sCity then
  begin
    bFound := True;
    tblMarathons.First;
    while NOT tblMarathons.Eof do
    begin
      if (tblMarathons['LocationID'] = tblLocations['LocationID'])
        AND (tblMarathons['Distance'] >= 40) then
      begin
        redQ2_2_3.Lines.Add(tblMarathons['MarathonName'] + #9 +
          FloatToStr(tblMarathons['Distance']));
      end;
      tblMarathons.Next;
    end;
  end;
  tblLocations.Next;
end;

if bFound = False then
  redQ2_2_3.Lines.Add(sCity + ' is not found.')
```

end;

```

// =====
// {$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}
// =====
procedure TfrmQuestion2.bmbRestoreDBClick(Sender: TObject);
begin
  // Restores the Database
  dbCONN.RestoreDatabase;
  redQ2_2_2.Clear;
  dbCONN.SetupGrids(dbgLocations, dbgMarathons, dbgrdSQL);
end;

procedure TfrmQuestion2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  // Disconnects from database and closes all open connections
```

```
    dbCONN.dbDisconnect;
end;

procedure TfrmQuestion2.FormCreate(Sender: TObject);
begin
    // Provided code
    redQ2_2_2.Paragraph.TabCount := 2;
    redQ2_2_2.Paragraph.Tab[0] := 150;
    redQ2_2_2.Paragraph.Tab[1] := 300;
end;

procedure TfrmQuestion2.FormShow(Sender: TObject);
begin
    // Sets up the connection to database and opens the tables.
    dbCONN := TConnection.Create;
    dbCONN.dbConnect;
    tblLocations := dbCONN.tblOne;
    tblMarathons := dbCONN.tblMany;
    dbCONN.SetupGrids(dbgLocations, dbgMarathons, dbgrdSQL);
    pgcDBAdmin.ActivePageIndex := 0;
end;

// =====
// {$ENDREGION}
// =====

end.
```

BYLAE G: OPLOSSING VIR VRAAG 3**Objekklas:**

```
unit MRecord_U;

interface

type
  TRecord = class(TObject)
  private
  var
    fMarathonName: String;
    fRecordHolder: String;
    fRecordDate: String;
    fRecordTime: String;
    fDistance: real;
  public
    // Provide code
    constructor create(sMarathonName, sRecordHolder, sRecordDate,
      sRecordTime: String; rDistance: real);
    procedure setRecordHolder(sName: String);
    procedure setRecordTime(sNewRecord: String);
    procedure setRecordDate(sNewDate: String);
    function toMinutes(sTime: String): real;

    function getRecordTime: String;
    function checkRecord(sRecordTime: String): boolean;
    function calcPace: real;
    function toString: String;
  end;

implementation

uses
  SysUtils, Math;

{ TRecord }

// =====
// 3.1.1 Constructor Create 3 punte
// =====

constructor TRecord.create(sMarathonName, sRecordHolder, sRecordDate,
  sRecordTime: String; rDistance: real);
begin
  fMarathonName := sMarathonName;
  fRecordHolder := sRecordHolder;
  fDistance := rDistance;
  fRecordDate := sRecordDate;
  fRecordTime := sRecordTime;
end;
```

```
// =====  
// 3.1.2 Function getRecordTime                                2 punte  
// =====  
  
function TMRecord.getRecordTime: String;  
begin  
    Result := fRecordTime;  
end;  
  
// =====  
// 3.1.3 Function checkRecord                                5 punte  
// =====  
  
function TMRecord.checkRecord(sRecordTime: String): boolean;  
begin  
    Result := sRecordTime < fRecordTime;  
    { Alternative:  
      if sRecordTime < fRecordTime then  
        Result := True  
      Else  
        Result := False;  
    }  
end;  
  
// =====  
// 3.1.4 Function calcPace                                  4 punte  
// =====  
  
function TMRecord.calcPace: real;  
begin  
    Result := toMinutes(fRecordTime) / fDistance;  
end;  
  
// =====  
// 3.1.5 Function toString                                  5 punte  
// =====  
  
function TMRecord.toString: String;  
begin  
    Result := fMarathonName + ' - ' + FloatToStr(fDistance)  
        + ' km: ' + fRecordHolder + ' (' + fRecordTime + ' on ' +  
        fRecordDate + ')';  
end;  
  
// =====  
// Provided code  
// =====  
procedure TMRecord.setRecordHolder(sName: String);  
begin  
    fRecordHolder := sName;  
end;  
  
procedure TMRecord.setRecordTime(sNewRecord: String);  
begin  
    fRecordTime := sNewRecord;  
end;
```

```
procedure TMRecord.setRecordDate(sNewDate: String);
begin
    fRecordDate := sNewDate;
end;
function TMRecord.toMinutes(sTime: String): real;
var
    rMin: real;
begin
    rMin := StrTofloat(copy(sTime, 4, 2));
    rMin := rMin + StrToFloat(copy(sTime, 1, 2)) * 60;
    rMin := rMin + StrToFloat(copy(sTime, 7, 2)) / 60;
    Result := rMin;
end;

end.
```

Hoofvormeenheid:

```
unit Question3_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, CheckLst, ExtCtrls, Buttons, Spin, ComCtrls,
  jpeg, pngimage;

type
  TfrmQuestion3 = class(TForm)
    gbxQ3_2_1: TGroupBox;
    gbxQ3_2_2: TGroupBox;
    redQ3: TRichEdit;
    btnQ3_2_1: TButton;
    gbxQ3_2_3: TGroupBox;
    btnQ3_2_3: TButton;
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ3_2_2: TButton;
    Label6: TLabel;
    edtQ3_2_1_Marathon: TEdit;
    Label2: TLabel;
    Label1: TLabel;
    edtQ3_2_1_RecordHolder: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    edtQ3_2_1_RecordTime: TEdit;
    rgpQ3_2_1: TRadioGroup;
    Label7: TLabel;
    edtQ3_2_3_Name: TEdit;
    edtQ3_2_3_Time: TEdit;
    Image1: TImage;
    edtQ3_2_1_RecordDate: TEdit;
    procedure btnQ3_2_1Click(Sender: TObject);
    procedure btnQ3_2_3Click(Sender: TObject);
    procedure btnQ3_2_2Click(Sender: TObject);
  private
  public
  end;
var
  frmQuestion3: TfrmQuestion3;
implementation
{$R *.dfm}
uses
  MRecord_U;

var
  objMRecord: TMRecord;
```

```
// =====  
// 3.2.1 Instantiate object 8 punte  
// =====
```

```
procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);  
var  
    sMarathonName, sRecordHolder, sRecordDate, sRecordTime: String;  
    rDistance: real;  
    sDistance: String;  
begin  
    // Provided code  
    redQ3.Clear;  
    sMarathonName := edtQ3_2_1_Marathon.Text;  
    sRecordHolder := edtQ3_2_1_RecordHolder.Text;  
    sRecordDate := edtQ3_2_1_RecordDate.Text;  
    sRecordTime := edtQ3_2_1_RecordTime.Text;  
  
    // Question 3.2.1  
  
    sDistance := rgpQ3_2_1.Items[rgpQ3_2_1.ItemIndex];  
    rDistance := StrToFloat(Copy(sDistance, 1, Pos(' ', sDistance) - 1));  
    objMRecord := TMRecord.create(sMarathonName, sRecordHolder,  
        sRecordDate, sRecordTime, rDistance);  
    redQ3.lines.Add(objMRecord.toString);  
end;
```

```
// =====  
// 3.2.2 Pace 4 punte  
// =====
```

```
procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);  
begin  
    // Provided code  
    redQ3.Clear;  
  
    // Question 3.2.2  
    redQ3.lines.Add('Record holder's pace:'+  
        FloatToStrF(objMRecord.calcPace, ffFixed, 8, 3) + '  
        min/km');  
end;
```

```
// =====  
// 3.2.3 Check record 9 punte  
// =====
```

```
procedure TfrmQuestion3.btnQ3_2_3Click(Sender: TObject);  
var  
    sName, sTime: String;  
begin  
    // Provided code  
    redQ3.Clear;  
  
    // Question 3.2.3  
    sName := edtQ3_2_3_Name.Text;  
    sTime := edtQ3_2_3_Time.Text;
```

```
if objMRecord.checkRecord(sTime) then
  begin
    objMRecord.setRecordHolderName(sName);
    objMRecord.setRecordTime(sTime);
    objMRecord.setRecordDate(DateToStr(Date()));
    redQ3.lines.Add(objMRecord.toString);
  end
Else
  begin
    redQ3.lines.Add
      ('The current record remains: ' + objMRecord.getRecordTime);
  end;
end;

end.
```

BYLAE H: OPLOSSING VIR VRAAG 4

```
unit Question4_u;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms, Dialogs, ExtCtrls, StdCtrls, ComCtrls, Buttons, pngimage;

type
    TfrmQ4_1 = class(TForm)
        pgcQ4: TPageControl;
        tshQ4_1: TTabSheet;
        tshQ4_2: TTabSheet;
        redQ4_1: TRichEdit;
        btnQ4_1: TButton;
        pnlQ4_1Heading: TPanel;
        redQ4_2: TRichEdit;
        cmbQ4_2: TComboBox;
        pnlQ4_2Heading: TPanel;
        btnReset: TBitBtn;
        memQ4_2: TMemo;
        imgQ4_1: TImage;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        procedure btnQ4_1Click(Sender: TObject);
        procedure display2D;
        procedure FormCreate(Sender: TObject);
        procedure cmbQ4_2Change(Sender: TObject);
        procedure btnResetClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    frmQ4_1: TfrmQ4_1;
    iCheck: Integer = 0;
    arrMarathons: array [1 .. 10] of String = (
        'Wally Hayward Marathon',
        'Sasol Marathon',
        'Soweto Marathon',
        'Jacaranda City Marathon',
        'Sasol Marathon',
        'Durban City Marathon',
        'Soweto Marathon',
        'Soweto Marathon',
        'Wally Hayward Marathon',
        'Soweto Marathon'
    );
```

```

arrChar: array [1 .. 14, 1 .. 14] of char =
  (('u', 'x', 'v', 'm', 's', 'a', 's', 'o', 'l', 'f', 'k', 'j', 't', 'r'),
  ('u', 'm', 'g', 'e', 'n', 'i', 'w', 'a', 't', 'e', 'r', 'd', 's', 'e'),
  ('g', 'v', 'o', 'e', 't', 'v', 'a', 'n', 'a', 'f', 'r', 'i', 'k', 'a'),
  ('e', 'p', 'o', 'y', 'i', 'l', 'c', 'k', 'h', 'j', 's', 'd', 'f', 'd'),
  ('n', 'k', 'n', 'y', 's', 'n', 'a', 'f', 'o', 'r', 'e', 's', 't', 'u'),
  ('i', 's', 'y', 'd', 'b', 'c', 'r', 'g', 'h', 'k', 'c', 's', 'a', 'r'),
  ('w', 'a', 'l', 'l', 'y', 'h', 'a', 'y', 'w', 'a', 'r', 'd', 's', 'b'),
  ('a', 's', 'q', 'r', 't', 'n', 'n', 'j', 'h', 'e', 'r', 't', 'h', 'a'),
  ('t', 'o', 'e', 'r', 'y', 'b', 'd', 'r', 'h', 'k', 'l', 'g', 'd', 'n'),
  ('e', 'j', 'a', 'c', 'a', 'r', 'a', 'n', 'd', 'a', 'c', 'i', 't', 'y'),
  ('r', 'y', 'j', 'f', 'g', 'f', 'c', 'f', 'g', 'u', 'h', 'v', 'c', 'i'),
  ('k', 'h', 'h', 'l', 'p', 'h', 'i', 'l', 'l', 'c', 'r', 'e', 's', 't'),
  ('a', 'd', 'e', 'v', 'd', 's', 'o', 'w', 'e', 't', 'o', 'm', 'k', 'y'),
  ('p', 'd', 'u', 'r', 'b', 'a', 'n', 'c', 'i', 't', 'y', 'z', 'c', 'l'));

```

implementation

```
{$R *.dfm}
```

```

// =====
// 4.1 Count marathons 11 punte
// =====

```

```

procedure TfrmQ4_1.btnQ4_1Click(Sender: TObject);
var
  iOut, iIn, iNumMarathons: Integer;
begin
  for iOut := 1 to 10 do
  begin
    iNumMarathons := 1;
    for iIn := iOut + 1 to 10 do
    begin
      if (arrMarathons[iOut] = arrMarathons[iIn]) and
          (arrMarathons[iIn] <> '') then
        begin
          inc(iNumMarathons);
          arrMarathons[iIn] := '';
        end;
      end;
    if arrMarathons[iOut] <> '' then
      redQ4_1.Lines.Add(arrMarathons[iOut] + #9 +
        IntToStr(iNumMarathons));
    end;
  end;
end;

```

```

// =====
// 4.2 Find hidden marathons 19 punte
// =====

procedure TfrmQ4_1.cmbQ4_2Change(Sender: TObject);
var
  iOut, iIn, iLen, iR, iC, iC2, iPos: Integer;
  sWord, sNewWord1, sNewWord2: String;
  bFound: boolean;
begin
  bFound := false;
  sWord := cmbQ4_2.Text;
  iLen := Length(sWord);
  for iOut := 1 to 14 do
  begin
    for iIn := 1 to 14 do
    begin
      iR := iOut;
      iC := iIn;
      if sWord[i] = arrChar[iOut, iIn] then
      begin

        sNewWord2 := '';
        while (iC <= 14) AND (bFound = false) do
        begin
          sNewWord2 := sNewWord2 + arrChar[iR, iC];

          if sNewWord2 = sWord then
          begin
            bFound := true;
            memQ4_2.Text := 'Row ' + IntToStr(iR)
              + '@ column ' + IntToStr(iIn) + ' to ' + IntToStr
                (iIn + iLen - 1); // iC - iLen + 1);
            for iC2 := iC downto iC - iLen + 1 do
              arrChar[iR, iC2] := upCase(arrChar[iR, iC2]);
            display2D;
          end;
          inc(iC);
        end;
      end;
    end;
  end;
end;

//=====
// Provided code - Do not change
//=====
=
procedure TfrmQ4_1.display2D;
var
  iOut, iIn: Integer;
  sOut: string;
begin
  redQ4_2.Clear;
  for iOut := 1 to 14 do
  begin

```

```
    sOut := sOut + #9 + IntToStr(iOut);
end;
redQ4_2.Lines.Add('' + #9 + sOut + #13);
for iOut := 1 to 14 do
begin
    sOut := #13 + IntToStr(iOut) + #9;
    for iIn := 1 to 14 do
    begin
        if iCheck = 1 then
        Begin
            arrChar[iOut, iIn] := lowercase(arrChar[iOut, iIn] + '')[1];
            sOut := sOut + #9 + arrChar[iOut, iIn];
        End
        else
            sOut := sOut + #9 + arrChar[iOut, iIn];
        end;
        redQ4_2.Lines.Add(sOut);
    end;
    iCheck := 0;
end;
procedure TfrmQ4_1.FormCreate(Sender: TObject);
var
    iOut, iIn: Integer;
    sOut: String;
begin
    pgcQ4.ActivePageIndex := 0;
    // Q4.2
    redQ4_2.Clear;
    redQ4_2.Paragraph.TabCount := 15;
    redQ4_2.Paragraph.Tab[0] := 20;
    redQ4_2.Paragraph.Tab[1] := 40;
    redQ4_2.Paragraph.Tab[2] := 60;
    redQ4_2.Paragraph.Tab[3] := 80;
    redQ4_2.Paragraph.Tab[4] := 100;
    redQ4_2.Paragraph.Tab[5] := 120;
    redQ4_2.Paragraph.Tab[6] := 140;
    redQ4_2.Paragraph.Tab[7] := 160;
    redQ4_2.Paragraph.Tab[8] := 180;
    redQ4_2.Paragraph.Tab[9] := 200;
    redQ4_2.Paragraph.Tab[10] := 220;
    redQ4_2.Paragraph.Tab[11] := 240;
    redQ4_2.Paragraph.Tab[12] := 260;
    redQ4_2.Paragraph.Tab[13] := 280;
    redQ4_2.Paragraph.Tab[14] := 300;
    display2D;
end;

procedure TfrmQ4_1.btnResetClick(Sender: TObject);
Var
    i: Integer;
begin
    iCheck := 1;
    display2D;
end;

end.
```