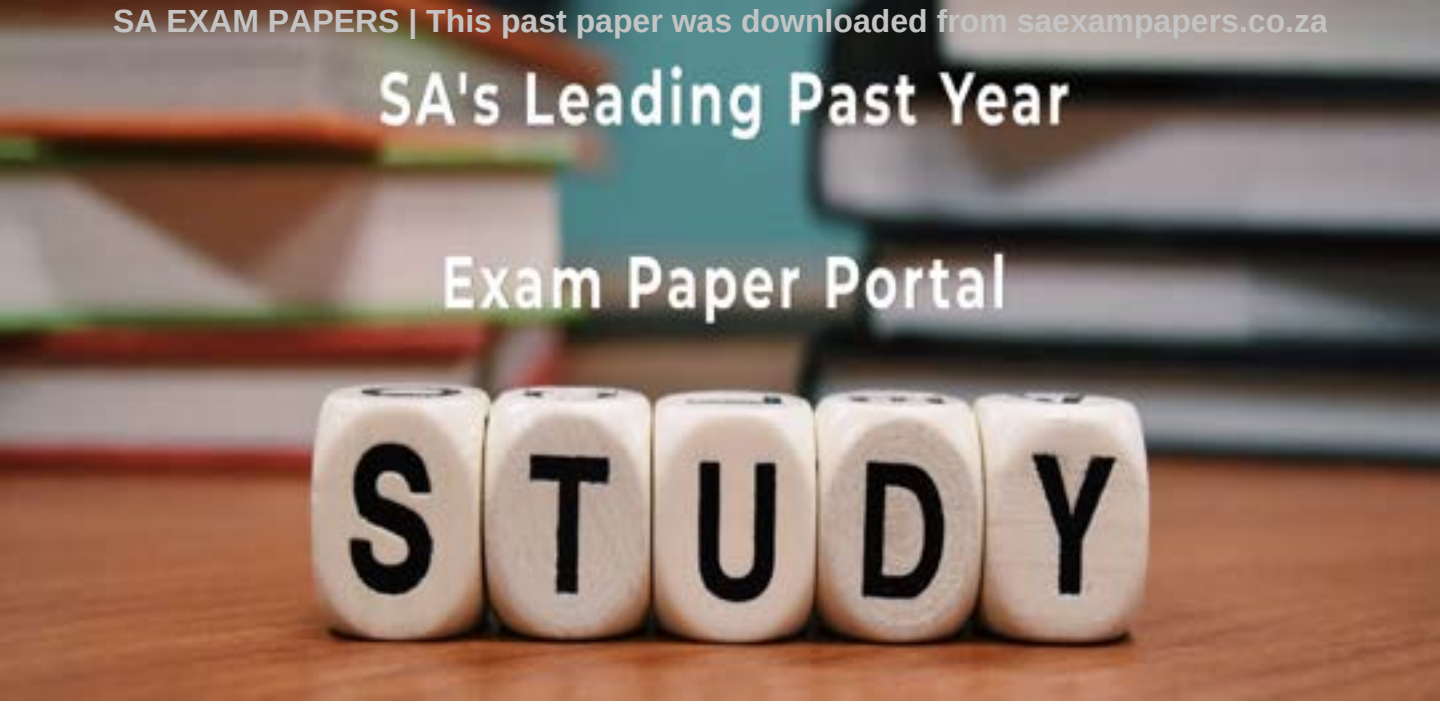

You have Downloaded, yet Another Great Resource to assist you with your Studies ☺

Thank You for Supporting SA Exam Papers

Your Leading Past Year Exam Paper Resource Portal

Visit us @ www.saexampapers.co.za

basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE 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	**Knoppie [1.1 - Random]** Genereer ewekansige getal ✓ van 5 tot 10 (beide waardes ingesluit in reeks) ✓ `edtQ1_1_1.Text :=` ✓ skakel getal om na String ✓	**4**	

1.1.2	**Knoppie [1.1.2 - Round up]**
	
	`spnQ1_1_2.Value :=` ✓
	Opgerond ✓
	`Ceil (NUMBER)`
	`Round (NUMBER + 0.5)`
	`Floor (NUMBER) + 1`
	`Trunc (NUMBER) + 1`
	Met konstante as parameter ✓
		3	

1.2	**Knoppie [1.2 - Surface area]**
	
	Onttrek die hoogte en radius✓ uit die redigeerblokkies
	Omgeskakel na Float ✓
	Bereken die oppervlakte
	A := PI * rR *✓ (rR + ✓ sqrt (✓ sqr(rR) + sqr(rH)✓))
	
	Vertoon die oppervlakte in die etiket ✓
	geformatteer tot 2 desimale plekke ✓
	`FormatFloat('0.00', A);`
	`FloatToStrF(A, ffFixed, 10, 2);`
	`Format('%.f',[A]);`
	
	AANVAAR ook `Power(rR,2)` in plaas van `Sqr`
	
	LET WEL: Hakkies moet korrek ingevoeg wees in die berekening.
		8	

1.3	**Knoppie [1.3 - Read file]** Verklaar veranderlikes vir die adres en slaapkamers ✓ `AssignFile(tFile, 'Houses.txt')` ✓ `Reset(tFile)` ✓ Lus deur die tekslêer met die regte voorwwarde ✓ 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' ✓ **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)	**9**	

1.4	**Knoppie [1.4 - Add name]**
	
	Onttrek die naam uit die kombinasielys ✓
	
	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 ✓
	
	Verwyder die geselekteerde naam van die kombinasielys ✓
	`cmbQ1_4.DeleteSelected`
	`cmbQ1_4.Items.Delete(cmbQ1_4.ItemIndex)`
	
	LET WEL:
	GEEN punt indien die item in die kombinasielys vervang word met 'n leë string en nie verwyder word nie.
		7	

1.5	**Knoppie [1.5 - Replace]** Lus deur die string ✓ Toets of karakter NIE 'n spasie is NIE✓ Voeg karakter by wagwoord-veranderlike in omgekeerde volgorde✓ Vertoon wagwoord in memo ✓ 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✓ Vertoon opgedateerde wagwoord in memo ✓ **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) **In die wagwoord gedeelte AANVAAR ook:** • 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:** • `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:** • + 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]**
	`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`✓
	
	AANVAAR ook:
	• `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]** 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	**5**	
2.2.2	**Knoppie [2.2.2 - Qualifying events]** 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) Toets vlag / teller ✓ Vertoon boodskap wat aandui dat die stad nie gevind is nie ✓ **LET WEL:**** • 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	**Konstruktor Create** 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 ✓	**3**	
3.1.2	**Funksie getRecordTime** Opskrif van funksie met string as terugstuur datatipe ✓ Stuur fRecordTime terug ✓	**2**	
3.1.3	**Funksie checkRecord** Opskrif van funksie met Boolean terugstuur datatipe ✓met string parameter✓ Toets of parameterwaarde < fRecordTime ✓ Stuur true terug ✓ Anders Stuur false terug ✓ **AANVAAR ook**: • StrToTime and toMinutes to compare the times. • Toets of fRecordTime > parameterwaarde	**5**	
3.1.4	**Funksie calcPace** Opskrif van funksie met reële terugstuurtipe ✓ Stuur terug ✓ toMinute(fRecordTime) ✓ / fDistance ✓	**4**	
3.1.5	**Funksie toString** met string terugstuurtipe Opskrif van funksie met string as terugstuurtipe ✓ Bou string met streep, 'km:', 'on' en hakkies ✓ Afstand omgeskakel na string ✓ Bevat al die attribute ✓ Stuur die string terug ✓	**5**	
	Subtotaal: Objekklas	**19**	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

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

BYLAE D

VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING

SENTRUMNOMMER:	**EKSAMENNOMMER:**		
VRAAG	**BESKRYWING**	**MAKS. PUNTE**	**LEERDER-PUNT**
4.1	**Knoppie [4.1 - Count marathons]** Buitenste lus van 1 tot 9 ✓ Inisialiseer teller✓ Binneste lus (buite + 1 tot 10) ✓ Toets array[buite] ✓= array[binne] ✓ en array[binne] <> simbool ✓ Inkrementeer teller ✓ vervang array[binne] met simbool soos bv ('*', ' ')✓ Indien array[buite] is NIE simbool wat gebruik is ✓ Vertoon array[buite] ✓ en teller omgeskakel na String✓ **Konsep 1: (Vervang duplikate wanneer tel)** Stap met lus deur skikking (1) Inisialiseer teller (1) Lus begin by buitelus-teller + 1 (1) Toets array[buite] (1) = array[binne] (1) EN array[binne] <> simbool (1) Inkrementeer teller (1) Vervang met array[binne] met spasie of * of ander simbool (1) Indien array[buite] is NIE simbool wat gebruik is (1) Vertoon array[buite] (1) en teller omgeskakel na string (1)	**11**	

	Konsep 2: (Skep lus (array/string) sonder duplikate) **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) **Lus deur unieke waardes in Temp skikking en tel in arrMarathons skikking [5]** 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) **Vertoon [2]** Vertoon marathon se naam (1) en teller (1) **Konsep 3: (Tydelike skikkings met unieke marathon name en teller waardes)** Gebruik `arrTempMarathons` en `arrCountMarathons` 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] ## Lus van 1 tot teller (1) Vertoon arrTempMarathons en arrCountMarathons (omgeskakel na heelgetal) (1)		

4.2	**Combobox [cmbQ4_2]** *Onttrek woord uit kombinasielys* ***[1]*** Lees woord uit kombinasielys✓ *Toets of woord in die ry voorkom* ***[10]*** Lus deur rye 1 tot 14 ✓ Inisialiseer nuweString ✓ Lus kolomme 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✓ *Vertoon die ry-nommer, begin- en eindkolom-indeks* ***[3]*** Vertoon ry-nommer ✓ beginkolom ✓ en eindkolom ✓ *Verander na hoofletters* ***[4]*** Lus deur woord in 2D skikking beginkolom-indeks ✓ Eindkolom-indeks ✓ Verander karakter in 2D ✓ na hoofletter-karakter ✓ *Vertoon 2D skikking* ***[1]*** Roep vertoon metode ✓	**19**	

	Konsep 1: (Ry- en Kolomlusse – kopieër uit ry) *Onttrek woord uit kombinasielys* ***[1]*** Lees woord uit kombinasielys (1) *Toets of woord in die ry voorkom* ***[10]*** 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) *Vertoon die ry-nommer, begin- en eindkolom-indeks* ***[3]*** Vertoon ry-nommer (1) beginkolom (1) en eindkolom (1) *Verander na hoofletters* ***[4]*** Lus van beginkolom-indeks (1) tot eindkolom (1) Verander karakter (1) na hoofletter-karakter (1) *Vertoon 2D skikking* ***[1]*** Roep vertoon metode (1) **Konsep 2: (Ry – gebruik pos direk)** *Onttrek woord uit kombinasielys* ***[1]*** Lees woord uit kombinasielys (1) *Toets of woord in die ry voorkom* ***[10]*** 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) *Vertoon die ry-nommer, begin- en eindkolom-indeks* ***[3]*** Vertoon ry-nommer (1) beginkolom (1) en eindkolom (1) *Verander na hoofletters* ***[4]*** Lus van beginkolom (1) tot eindkolom (1) Verander karakter (1) na hoofletter-karakter (1) *Vertoon 2D skikking* ***[1]*** Roep vertoon metode (1)		
	TOTAAL AFDELING D:	**30**	

OPSOMMING VAN LEERDER SE PUNTE:

SENTRUMNOMMER:		**LEERDER SE EKSAMENNOMMER:**			
	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
```

```
{$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;

// ========================================================================
// 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
  TMRecord = 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;

{ TMRecord }

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

constructor TMRecord.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;
```

```
// ============================================================================
// 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[1] = 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.
```