



# basic education

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

## **NASIONALE SENIOR SERTIFIKAAT**

**GRAAD 12**

**INLIGTINGSTEGNOLOGIE V1**

**NOVEMBER 2019**

**NASIENRIGLYNE**

**PUNTE: 150**

**Hierdie nasienriglyne bestaan uit 24 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 na 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 kandidate 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.
- **Bylaag A, B, C en D** (bladsy 3-12) sluit die nasienrubriek vir elke om te gebruik vir enigeen van die twee programmeringstale in.
- **Bylaag E, F, G en H** (bladsy 13-24) bevat voorbeelde in programmeringskode van oplossings vir Vraag 1 tot Vraag 4.
- Kopieë van **Bylaag A, B, C, D en die opsomming van die leerder se punte** (bladsy 3-12) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

**BYLAAG A****VRAAG 1: NASIENRUBRIEK - ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1	<p><b>Knoppie [1.1 – Vertoon bedrag verskuldig]</b></p> <p>Verklaar 'n konstante veranderlike PRYS en ken waarde 14.95 toe✓            Verklaar 'n reële veranderlike vir die bedrag verskuldig✓            Verklaar 'n veranderlike vir die aantal pizzas✓            Kry getal uit spnV1_1✓            Bereken bedrag deur PRYS te gebruik✓            Verander skrifgrootte van byskrif ('label') na 20pt ✓            Vertoon bedrag op lblV1_1✓ omgeskakel na string en geldeenheid-formaat ✓</p>	8	
1.2	<p><b>Knoppie [1.2 – Pythagoras]</b></p> <p>Skep veranderlikes vir sye A, B en C✓            Ken lengte van 4 toe aan A ✓            Lengte vir sy B ✓ en skakel om na getal formaat✓  <math>C = \text{sqr}(\text{sqr}(A) + \text{sqr}(B))</math>                korrekte gebruik van <math>\text{sqr}(A)</math> OF <math>A * A</math>✓                korrekte gebruik van <math>\text{sqr}(rB)</math> OF <math>B * B</math>✓                kry die som van die twee waardes✓            Vertoon lengte van sy C op paneel✓            geformateer tot een desimale plek ✓</p> <p><b>Aanvaar:</b>            Sqr: Power(A,2)            Sqrt: Power((sqr(A) +sqr(B)),0.5)</p>	10	
1.3	<p><b>Knoppie [1.3 – Bepaal kleinste getal]</b></p> <p>Ken ewekansige getal ✓in die korrekte gebied ✓ aan veranderlike toe            Vertoon getal✓ omgeskakel na string✓            As (if) ✓getal &lt; kleinste ✓                Ken getal toe aan kleinste ✓            Vertoon kleinste in edtV1_3✓ omgeskakel na string✓</p> <p>RandomRange(1,101)            Random(100) + 1            Ceil(Random() *100)</p>	9	

1.4	<p><b>Knoppie [1.4 – Vertoon gedekripteerde string]</b></p> <p>Lees toevoerstring met 'n toevoerdialoogblokkie in ✓ met korrekte parameters ✓</p> <p>Lus ('Loop') ✓ van eerste ✓ tot laaste karakter ✓</p> <p>Korrekte toets uitgevoer (if/case) ✓</p> <p>Verkry karakter by indeks ✓</p> <p>Toets teen syfer ✓ vir vir al 10 syfers ✓</p> <p>Vervang syfer by korrekte indeks ✓</p> <p>Met korrekte karakter ✓ vir al 10 karakters ✓</p> <p>Vertoon ✓</p> <p><b>KONSEPTE:</b></p> <p>Lees toevoerstring met 'n toevoerdialoogblokkie(1) met korrekte parameters (1)</p> <p>Lus ('Loop') (1) van eerste (1) tot laaste karakter (1)</p> <p>Korrekte toets (if/case) (1)</p> <p>Onttrek karakter by indeks (1)</p> <p>Toets teen syfer (1) vir al 10 syfers (1)</p> <p>Vervang syfer by korrekte indeks (1)</p> <p>Met korrekte karakter (1) vir al 10 karakters (1)</p> <p>Vertoon (1)</p>	13	
	<b>TOTAAL AFDELING A:</b>	<b>40</b>	

**BYLAAG B****VRAAG 2: NASIENRUBRIEK – DATABASISPROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	<b>SQL-stellings</b>		
2.1.1	<b>Knoppie [2.1.1 – Beste speler]</b> SELECT SpelerVan, SpelerNaam FROM tblSpelers WHERE VaardigheidsVlak = 10 <b>KONSEPTE:</b> SELECT SpelerVan, SpelerNaam ✓ FROM tblSpelers ✓ WHERE VaardigheidsVlak = 10 ✓	3	
2.1.2	<b>Knoppie [2.1.2 – B-span-afrigters]</b> SELECT Afrigter, SpanNaam FROM tblSpanne WHERE SpanNaam Like "%B" <b>KONSEPTE:</b> SELECT Afrigter, SpanNaam ✓ FROM tblSpanne ✓ WHERE SpanNaam LIKE ✓ "%B" ✓	4	
2.1.3	<b>Knoppie [2.1.3 – Persentasie wedstryde gewen]</b> SELECT SpanNaam, Afrigter, (GetalWedstrydeGewen/GetalWedstrydeGespeel*100) AS [PersentasieWedstrydeGewen] FROM tblSpanne WHERE SpanNaam = ' ' + sSpan + ' ' <b>KONSEPTE:</b> SELECT SpanNaam, Afrigter, ✓ (GetalWedstrydeGewen/GetalWedstrydeGespeel ✓*100), ✓ AS [PersentasieWedstrydeGewen] FROM tblSpanne WHERE SpanNaam = ' ' + sSpan + ' ' ✓ <b>ALTERNATIEF:</b> 'WHERE SpanNaam = ' + QuotedStr(sSpan)	4	

2.1.4	<b>Knoppie [2.1.4 – Spangemiddeld meer as 6]</b>	5	
	<pre>SELECT SpanNaam, ROUND(AVG(VaardigheidsVlak),1) AS [GemidVaardigheidsVlak] FROM tblSpelers GROUP BY SpanNaam HAVING AVG(VaardigheidsVlak)&gt;6</pre> <p><b>KONSEPTE:</b>  <pre>SELECT SpanNaam, ROUND(AVG(VaardigheidsVlak) ✓,1)✓ AS GemidVaardigheidsVlak ✓ FROM tblSpelers GROUP BY SpanNaam✓ HAVING AVG(VaardigheidsVlak) &gt; 6✓</pre></p>		
2.1.5	<b>Knoppie [2.1.5 – Dateer wedstryde gewen op]</b>	3	
	<pre>UPDATE tblSpanne SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1 WHERE SpanNaam &lt;&gt; "o/14 B"</pre> <p><b>KONSEPTE:</b>  <pre>UPDATE tblSpanne ✓ SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1 ✓ WHERE SpanNaam &lt;&gt; "o/14 B"✓</pre></p>		
	<b>Subtotaal:</b>	<b>19</b>	

2.2	<b>Databasismanipulasie deur Delphi-kode te gebruik</b>		
2.2.1	<b>Knoppie [2.2.1 – Junior spelers in die o/18 A-span]</b> AssignFile ✓ & Rewrite ✓ Stel tblSpelers om by eerste rekord te begin lees ✓ Lus ('Loop') terwyl NOT tblSpelers.EOF ✓ Toets of SpanNaam 'o/18 A' is ✓ EN of eerste 2 karakters van IDnommer-veld >=3 is ✓ Skryf van en naam na lêer ✓ Vermeerder teller met 1 ✓ Gaan na volgende rekord in tblSpelers ✓ Eindig lus Maak lêer toe ✓ Vertoon teller van getal junior speler in byskrif ('label') ✓	11	

2.2.2	<p><b>Knoppie [2.2.2 – Afrigter en doelwagter se inligting]</b></p> <p>Lus ('Loop') terwyl NOT tblSpanne.EOF✓          Stel tblSpelers om te begin lees by eerste rekord✓          Lus ('Loop') terwyl NOT tblSpelers.EOF✓          Toets of              die SpanNaam-veld in tblSpanne gelyk is aan              SpanNaam-veld in tblSpelers ✓              EN DoelWagter = true✓              Voeg span se naam, afrigter se naam ✓              Speler se van en naam                  in regte formaat by afvoerstring✓          Gaan na volgende rekord in tblSpelers✓          Vertoon afvoer in reel✓          Gaan na volgende rekord in tblSpanne✓</p> <p><b>Alternatief:</b></p> <p>Lus ('Loop') terwyl NOT tblSpelers.EOF (1)          Toets of DoelWagter = true(1)          Stel tblSpanne om te begin lees by eerste rekord(1)          Lus ('Loop') terwyl NOT tblTeams.EOF(1)          Toets of die SpanNaam-veld in tblSpanne gelyk is aan              SpanNaam-veld in tblSpelers (1)              Voeg span se naam, afrigter se naam (1)              Speler se van en naam                  in regte formaat by afvoerstring(1)          Gaan na volgende rekord in tblSpanne(1)          Vertoon afvoer in reel(1)          Gaan na volgende rekord in tblSpelers(1)</p>	<b>10</b>	
	<b>Subtotaal:</b>	<b>21</b>	
	<b>TOTAAL AFDELING B:</b>	<b>40</b>	

**BYLAAG C****VRAAG 3: NASIENRUBRIEK - OBJEK-GEÛRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	<b>Konstruktor-metode:</b> Opskrif met regte parameters en datatipes ✓ Ken naam van speler parameterwaarde toe aan fSpelerNaam ✓ Ken gewig van speler parameterwaarde toe aan fGewigVanSpeler ✓ Stel fTelling op 0 ✓	4	
3.1.2	<b>getTelling-metode:</b> Funksie opskrif met heelgetal as terugstuur-datatipe ✓ fTelling aan 'result' toegeken ✓	2	
3.1.3	<b>dateerTellingOp-metode:</b> Prosedure-opskrif met heelgetalparameter ✓ Vermeerder fTelling ✓ met parameterwaarde ✓	3	
3.1.4	<b>berekenLMI-metode:</b> Funksie verklaar met reële terugstuur datatipe en ontvang reële parameterwaarde and return 'n berekende LMI ✓ Calculation = fGewigVanSpeler / ✓ Sqr(parameterwaarde lengte van speler) ✓	3	
3.1.5	<b>geskikVirSeleksie-metode:</b> Toets of telling 'n waarde tussen 0 en 7 is (inklusief) ✓ Result = Klein moontlikheid ✓ Toets of telling 'n waarde tussen 8 en 14 is (inklusief) Result = Gemiddelde moontlikheid ✓ Toets of telling > 14 is Result = Groot moontlikheid ✓	4	
3.1.6	<b>toString-metode:</b> Byskrifte (Naam, Gewig, Huidige telling) ✓ Regte attribute ✓ Regte omskakelings (gewig – float; telling – integer) ✓ Return string ✓	4	
	<b>Subtotaal: Objekklas</b>	<b>20</b>	



**VRAAG 3: NASIENRUBRIEK (VERVOLG)**

<b>VRAAG</b>	<b>BESKRYWING</b>	<b>MAKS. PUNTE</b>	<b>LEERDER - PUNT</b>
3.2.1	<p><b>Knoppie [3.2.1 - Instansieer speler-objek]</b></p> <p><i>Instansieer die objSpeler-objek:</i>  objSpeler ✓ := TSpeler.Create✓  Stuur naam en spelergewig in korrekte volgorde✓  met korrekte datatype (dieselfde as konstruktore) ✓  Gebruik 'n dialoogboks om aan te dui dat die objek ge-  instansieer is ✓</p>	<b>5</b>	
3.2.2	<p><b>Knoppie [3.2.2 – Bereken LMI]</b></p> <p>Gebruik toevoerdialoogblokkie ✓ om lengte in te sleutel  Roep berekenLMI-metode deur gebruik van objek✓  gebruik lengte as argument✓ omgeskakel na reële getal  Vertoon die inligting in <b>redV3_2_2</b> ✓  deur die toString-metode van die objek te gebruik ✓  Vertoon die LMI in <b>redV3_2_2</b> ✓  in die regte formaat met een desimale plek ✓</p>	<b>7</b>	
3.2.3	<p><b>Knoppie [3.2.3 – Dateer telling op]</b></p> <p>Kry itemindex uit die radio group✓  Om die telling te kry ✓  Roep die dateerTellingOp metode ✓  met korrekte argument✓  Vertoon opgedateerde telling op pnIV3_2_3 ✓  deur die getTelling-metode te gebruik ✓</p>	<b>6</b>	
3.2.4	<p><b>Knoppie [3.2.4 – Geskik vir seleksie]</b></p> <p>Roep die relevante metode as volg:  Vertoon op die byskrif ('label') <b>IbIV3_2_4</b>✓  Deur gebruik te maak van objSpeler.geskikVirSeleksie✓</p>	<b>2</b>	
	<b>Subtotaal: Form-klas</b>	<b>20</b>	
	<b>TOTAAL AFDELING C:</b>	<b>40</b>	

**VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
4.1	<b>Knoppie [4.1 – Vertoon Doolhof]</b> Kies doolhof uit kombinasieblokkie ✓ Regte lêernaam + '.txt' ✓ Toets en vertoon 'n boodskap indien die lêer nie bestaan nie ✓ Assign file, Reset die tekslêer ✓ Initialiseer indeks-veranderlike ✓ Lus (Loop) deur die tekslêer ✓ Inkrementeer die waarde van indeks ✓ (Posisie word bepaal deur beginwaarde) Lees reël ✓ ken die reël toe aan die skikking arrDoolhof ✓ Vertoon die reëlnommer ✓ en die reël ✓	11	

4.2	<p><b>Knoppie [4.2 – Langste gang]</b></p> <p>Inisialieer veranderlike na 0 om maksimum waarde te stoor✓  Lus (Loop) deur die skikking✓  Initialiseer veranderlike vir langste gang in die reël ✓  Initialiseer veranderlike om agtereenvolgende '-' in reël te tel✓  Lus (Loop) deur elke karakter✓ in die reël ✓  Toets of karakter 'n '-' is ✓  Inkrementeer dash-teller✓  As dash-teller meer is as langste gang in reël ✓  Vervang langste gang met dash-teller✓  Anders as karakter nie '-' is nie✓  Stel dash-teller na 0✓  Stoor langste gang in die ry in arrRye skikking/ 'n string✓  As langste in ry langer is as langste ✓ – vervang langste ✓  Vertoon boodskap met lengte van langste gang✓  Lus (Loop) deur skikking/string met langste gange per reël ✓  Vertoon reëlnommer van rye ✓  wat dieselfde as langste gang se lengte is✓</p> <p><b>KONSEPTE:</b>  Inisialiseer veranderlike vir langste gang na 0 (1)  Lus deur skikking (1)  Initialiseer dash-teller per reël(1)  en maksimum aantal aaneenlopende dashes per reël (1)  Lus (loop) deur elke karakter in die reël (2)  Toets of karakter dash is (1)  Vermeerder die dash-teller (1)  Toets of die dash-teller langste in die reël (2)  en stoor in maksimum aantal(1)  As karakter nie dash is nie (1)  Herstel dash-teller na 0 (1)</p> <p>Bepaal die langste gang in al die lyne in die doolhof  (Toets en stoor maksimum aantal) (2)  Vertoon langste (1)  Toets vir lyne wat selfde lengte het as langste gang (2)  Vertoon lyn nommer(s) (1)</p>	19	
	<b>TOTAAL AFDELING D:</b>	<b>30</b>	
	<b>GROOTTOTAAL:</b>	<b>150</b>	

**OPSOMMING VAN LEERDER SE PUNTE:**

<b>SENTRUMNOMMER:</b>		<b>LEERDER SE EKSAMENNOMMER:</b>			
	<b>AFDELING A</b>	<b>AFDELING B</b>	<b>AFDELING C</b>	<b>AFDELING D</b>	
	<b>VRAAG 1</b>	<b>VRAAG 2</b>	<b>VRAAG 3</b>	<b>VRAAG 4</b>	<b>GROOT-TOTAAL</b>
<b>MAKS. PUNTE</b>	<b>40</b>	<b>40</b>	<b>40</b>	<b>30</b>	<b>150</b>
<b>LEERDER SE PUNTE</b>					

**BYLAAG E: OPLOSSING VIR VRAAG 1**

```
unit Vraag1_u;

interface

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

type
  TfrmV1 = class(TForm)
    gpbV1_1: TGroupBox;
    gpbV1_4: TGroupBox;
    gpbV1_3: TGroupBox;
    btnV1_1: TButton;
    btnV1_4: TButton;
    edtV1_3: TEdit;
    btnV1_3: TButton;
    redV1_3: TRichEdit;
    btnReset: TButton;
    gpbV1_2: TGroupBox;
    btnV1_2: TButton;
    Label1: TLabel;
    pnlV1_2: TPanel;
    lblV1_1: TLabel;
    lbl1_2Pies: TLabel;
    edtV1_2Deursnee: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    edtV1_2Breedte: TEdit;
    Label4: TLabel;
    spnV1_1: TSpinEdit;
    Label5: TLabel;

    procedure btnResetClick(Sender: TObject);
    procedure btnV1_1Click(Sender: TObject);
    procedure btnV1_2Click(Sender: TObject);
    procedure btnV1_3Click(Sender: TObject);
    procedure btnV1_4Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);

  private
    { Private declarations }
  public
    iKleinste: integer; // global
    { Public declarations }
  end;

var
  frmV1: TfrmV1;

implementation

{$R *.dfm}
```

## =====

**Vraag 1.1 = 8 punte**

```
=====
procedure TfrmV1.btnV1_1Click(Sender: TObject);
const
  PRYS = 14.95;
var
  iGetal: integer;
  rBedrag: real;
begin
  lblV1_1.Font.Size := 20;
  iGetal := spnV1_1.Value;
  rBedrag := iGetal * PRYS;
  lblV1_1.Caption := FloatToStrF(rBedrag, ffCurrency, 10, 2);
end;
```

## =====

**Vraag 1.2 = 10 punte**

```
=====
procedure TfrmV1.btnV1_2Click(Sender: TObject);
var
  rA, rB, rC : real;
begin
  rA := 4;
  rB := StrToFloat(edtV1_2.Text);
  rC := Sqrt(Sqr(rA) + Sqr(rB));
  pnlV1_2.Caption := FloatToStrF(rC, ffFixed, 8, 1);
end;
```

## =====

**Vraag 1.3 = 9 punte**

```
=====
procedure TfrmV1.btnV1_3Click(Sender: TObject);
var // Kode wat voorsien is
  iGetal: integer;

begin
  iGetal := random(100) + 1;
  redV1_3.lines.Add(IntToStr(iGetal));
  if iGetal < iKleinste then
    iKleinste := iGetal;
  edtV1_3.Text := IntToStr(iKleinste);
end;
```

## =====

**Vraag 1.4 = 13 punte**

```
=====
procedure TfrmV1.btnV1_4Click(Sender: TObject);
const
  ALFA = 'ABCDEFGHJIJ';

var // Kode wat voorsien is
    sString, sVertoon: String;
    I, iIndeks: integer;

begin
  // Kode wat voorsien is
  sString := InputBox('Sleutel 'n geënkripteerde string in: ', '', '');
  sVertoon := '';
  for I := 1 to Length(sString) do
    if sString[I] in ['0' .. '9'] then
      begin
        iIndeks := StrToInt(sString[I]) + 1;
        sVertoon := sVertoon + ALFA[iIndeks];
      end
    else
      sVertoon := sVertoon + sString[I];
  ShowMessage(sVertoon);

end;
```

## =====

**Kode wat voorsien is**

```
=====
procedure TfrmV1.btnResetClick(Sender: TObject);
begin
  // Kode wat voorsien is - moenie verander nie
  iKleinste := 100;
  redV1_3.Clear;
  edtV1_3.Clear;
end;

procedure TfrmV1.FormActivate(Sender: TObject);
begin
  // Kode wat voorsien is - moenie verander nie
  iKleinste := 100;
  redV1_3.Clear;
  edtV1_3.Clear;
  CurrencyString := 'R';
end;

end.
```

**BYLAAG F: OPLOSSING VIR VRAAG 2****VRAAG 2.1: SQL kode**

```
=====
Vraag 2.1: 2.1.1(3), 2.1.2(4), 2.1.3(4), 2.1.4(5), 2.1.5 (3)
=====
```

```
2.1.1: SELECT SpelerVan, SpelerNaam
        FROM tblSpelers WHERE VaardigheidsVlak = 10

2.1.2: SELECT Afrigter, SpanNaam
        FROM tblSpanne
        WHERE SpanNaam Like "%B"

2.1.3: SELECT SpanNaam, Afrigter,
        (GetalWedstrydeGewen/GetalWedstrydeGespeel*100) AS
        [PersentasieWedstrydeGewen]
        FROM tblSpanne
        WHERE SpanNaam = "' + sSpan + '"

2.1.4: SELECT SpanNaam, round(avg(VaardigheidsVlak),1) AS
        [GemidVaardigheidsVlak]
        FROM tblSpelers
        GROUP BY SpanNaam HAVING avg(VaardigheidsVlak) > 6

2.1.5: UPDATE tblSpanne
        SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1
        WHERE SpanNaam <> "o/14 B"
```

**QUESTION 2.2: DATABASE MANIPULATION met gebruik van Delphi-kode**

```
// {$REGION 'VRAAG 2.2'}
```

```
=====
Vraag 2.2.1 = 11 punte
=====
```

```
procedure TfrmDBVraag2.btnV2_2_1Click(Sender: TObject);
var
    tFile: textfile;
    iTel, iJaar: integer;
begin
    AssignFile(tFile, 'Junior18A.txt');
    Rewrite(tFile);
    tblSpelers.first;
    iTel := 0;
    while not tblSpelers.eof do
    begin
        iJaar := StrToInt(copy(tblSpelers['IDNommer'], 1, 2));
        if (tblSpelers['SpanNaam'] = 'o/18 A') AND (iJaar >= 3) then
        begin
            Writeln(tFile,
                tblSpelers['SpelerVan'] + ' ' + tblSpelers['SpelerNaam']);
            inc(iTel);
        end;
        tblSpelers.Next;
    end;
end;
```



```

CloseFile(tFile);
lblV2_2_1.Caption := 'Getal junior spelers in die o/18 A-span: ' +
IntToStr(iTel);

// Kode wat voorsien is
dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

=====
Vraag 2.2.2 = 10 punte
=====
procedure TfrmDBVraag2.btnV2_2_2Click(Sender: TObject);
var
  sAfvoer: String; // veranderlike om span wat geselekteer is te stoor

begin // Vraag 2.2.2
  // Kode wat voorsien is
  redV2_2_2.Clear;
  redV2_2_2.Paragraph.TabCount := 2;
  redV2_2_2.Paragraph.Tab[0] := 70;
  redV2_2_2.Paragraph.Tab[1] := 160;
  redV2_2_2.Lines.Add('Span' + #9 + 'Afrigter' + #9 + 'Doelwagter');

  // -----
  // Sleutel jou kode hier in:

  tblSpanne.first;
  while NOT tblSpanne.eof do
  begin
    sAfvoer := tblSpanne['SpanNaam'] + #9 + tblSpanne['Afrigter'] + #9;
    tblSpelers.first;
    while NOT tblSpelers.eof do
    begin
      if (tblSpanne['SpanNaam'] = tblSpelers['SpanNaam']) AND
        (tblSpelers['DoelWagter'] = true) then
      begin
        sAfvoer := sAfvoer + tblSpelers['SpelerVan'] + ', ' + tblSpelers
          ['SpelerNaam'];
      end;
      tblSpelers.Next;
    end; tblSpanne.Next;
    redV2_2_2.Lines.Add(sAfvoer);
  end;

  // Kode wat voorsien is
  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

```

```
=====
{$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}
=====
procedure TfrmDBVraag2.bmbHerstelDBClick(Sender: TObject);
begin
  // Herstel die database
  dbCONN.RestoreDatabase;
  redV2_2_2.Clear;
  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

// =====
procedure TfrmDBVraag2.FormClose(Sender: TObject;

  var Action: TCloseAction);
begin // Diskonnekteer databasis en maak alle oop konneksies toe
  dbCONN.dbDisconnect;
end;

procedure TfrmDBVraag2.FormCreate(Sender: TObject);
begin
  CurrencyString := 'R';
end;

// =====
procedure TfrmDBVraag2.FormShow(Sender: TObject);
begin // Stel konneksie met databasis op en maak die tabelle oop.
  dbCONN := TConnection.Create;
  dbCONN.dbConnect;
  tblSpanne := dbCONN.tblOne;
  tblSpelers := dbCONN.tblMany;

  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
  pgcDBAdmin.ActivePageIndex := 0;
end;
// =====
// {$ENDREGION}

end.
```

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

```

unit Speler_U;

interface

uses StdCtrls, SysUtils;

type
  TSpeler = class(TObject)

    //Kode wat voorsien is - moenie verander nie
  private
    fSpelerNaam: String;
    fGewigVanSpeler : real;
    fTelling : integer;

  public
    //Kode wat voorsien is - moenie verander nie
    constructor create(sSpelerNaam : String; rGewigVanSpeler : real);
    function getTelling : integer;

    //Voeg kode hier in
    function berekenLMI (rLengteVanSpeler : real) : real;
    procedure dateerTellingOp (iTelling : integer);
    function geskikVirSeleksie: String;
    function toString : String;
end;

```

```

implementation

```

```

{ TSpeler }

```

```

=====
Vraag 3.1.1 = 4 punte
=====

```

```

constructor TSpeler.create(sSpelerNaam : String; rGewigVanSpeler :
real);
begin
  fSpelerNaam := sSpelerNaam;
  fGewigVanSpeler := rGewigVanSpeler;
  fTelling := 0;
end;

```

```

=====
Vraag 3.1.2 = 2 punte
=====

```

```

function TSpeler.getTelling: integer;
begin
  result := fTelling;
end;

```

```

=====
Vraag 3.1.3 = 3 punte
=====

```

```

procedure TSpeler.dateerTellingOp(iTelling: integer);
begin
  fTelling := fTelling + iTelling;
end;

```

=====  
**Vraag 3.1.4 = 3 punte**  
=====

```
function TSpeler.berekenLMI(rLengteVanSpeler: real) : real;
begin
  result := fGewigVanSpeler / sqr(rLengteVanSpeler);
end;
```

=====  
**Vraag 3.1.5 = 4 punte**  
=====

```
function TSpeler.geskikVirSeleksie:string;
begin
  if fTelling < 8 then
    result := 'Klein moontlikheid'
  else
    if fTelling < 15 then
      result := 'Gemiddelde moontlikheid'
    else
      result := 'Groot moontlikheid'
end;
```

=====  
**Vraag 3.1.6 = 4 punte**  
=====

```
function TSpeler.toString : String;
begin
  result := 'Naam: ' + fSpelerNaam + #13 + 'Gewig: '+
    FloatToStr(fGewigVanSpeler)+ #13+ 'Huidige telling: ' +
    IntToStr (fTelling);
end;

end.
```

**Hoofvormeenheid**

```
unit Vraag3_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Speler_U, ComCtrls, ExtCtrls, Spin, DateUtils;

type
  TfrmVraag3 = class(TForm)
    lblNaamVanSpeler: TLabel;
    edtNaamVanSpeler: TEdit;
    btnV3_2_2: TButton;
    redV3_2_2: TRichEdit;
    pnlV3_2_3: TPanel;
    btnV3_2_3: TButton;
    btnV3_2_1: TButton;
    rgpV3_2_3: TRadioGroup;
    btnV3_2_4: TButton;
    lblV3_2_4: TLabel;
    Label1: TLabel;
    edtGewigVanSpeler: TEdit;
    Label3: TLabel;
    grpV3_2_1: TGroupBox;
    grpV3_2_2: TGroupBox;
    grpV3_2_3: TGroupBox;
    grpV3_2_4: TGroupBox;
    procedure btnV3_2_2Click(Sender: TObject);
    procedure btnV3_2_3Click(Sender: TObject);
    procedure btnV3_2_1Click(Sender: TObject);
    procedure btnV3_2_4Click(Sender: TObject);
    // procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag3: TfrmVraag3;
  // Veranderlike wat reeds verklaar is
  objSpeler : TSpeler;
implementation

{$R *.dfm}
```

```
=====
Vraag 3.2.1 = 5 punte
=====
```

```
procedure TfrmVraag3.btnV3_2_1Click(Sender: TObject);
var
  sNaamVanSpeler : String;
  rGewigVanSpeler : real;
begin
  sNaamVanSpeler := edtNaamVanSpeler.Text; rGewigVanSpeler :=
  StrToFloat(edtGewigVanSpeler.Text); objSpeler :=
  TSpeler.create(sNaamVanSpeler,rGewigVanSpeler);
  ShowMessage('Speler-objek is ge-instansieer.');
```

```
=====
Vraag 3.2.2 = 7 punte
=====
```

```
procedure TfrmVraag3.btnV3_2_2Click(Sender: TObject);
var
  rLengteVanSpeler, rLMI : real;
  sLMI : String;

begin
  redV3_2_2.Clear;
  rLengteVanSpeler := StrToFloat(InputBox('Lengte van die speler',
  'Sleutel die lengte van die speler in meter in: ',''));
  rLMI := objSpeler.berekenLMI(rLengteVanSpeler);
  sLMI := FloatToStrF(rLMI,ffFixed,3,1);
  redV3_2_2.Lines.Add(objSpeler.toString);
  redV3_2_2.Lines.Add ('LMI: '+ sLMI);
end;
```

```
=====
Vraag 3.2.3 = 6 punte
=====
```

```
procedure TfrmVraag3.btnV3_2_3Click(Sender: TObject);
var
  iTelling: integer;
begin
  iTelling := StrToInt(rgpV3_2_3.Items[rgpV3_2_3.ItemIndex]);
  objSpeler.dateerTellingOp(iTelling);
  pnlV3_2_3.Caption :=('Opgedateerde telling: ' +
  IntToStr(objSpeler.getTelling));
end;
```

```
=====
Vraag 3.2.4 = 2 punte
=====
```

```
procedure TfrmVraag3.btnV3_2_4Click(Sender: TObject);
begin
  lblV3_2_4.Caption := objSpeler.geskikVirSeleksie;
end;
end.
```

**BYLAAG H: OPLOSSING VIR VRAAG 4**

```
unit Vraag4_U;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ComCtrls;
type
  TfrmV4 = class(TForm)
    gpbVrae: TGroupBox;
    btnV4_1: TButton;
    btnV4_2: TButton;
    cmbV4_1: TComboBox;
    Label1: TLabel;
    redV4: TRichEdit;
    procedure btnV4_1Click(Sender: TObject);
    procedure btnV4_2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  // Kode wat voorsien is
const
  iRyTel = 16;

var
  frmV4: TfrmV4;
  arrDoolhof: array [1..iRyTel] of String;
```

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

---

---

**Vraag 4.1 = 11 punte**

---

---

```
procedure TfrmV4.btnV4_1Click(Sender: TObject);
```

```
var
  iIndeks: integer;
  sLeerNaam, sReel: String;
  tNaamLeer: TextFile;
```

```
begin
```

---

---

**Kode wat voorsien is**

---

---

```
redV4.Clear;
```

---

---

**Kode wat voltooi moet word**

---

---

```
iIndeks := 1;
sLeerNaam := cmbV4_1.Text + '.txt';
if FileExists(sLeerNaam) then
begin
  AssignFile(tNaamLeer, sLeerNaam);
  Reset(tNaamLeer);
```

```

while not (eof(tNaamLeer)) do
begin
  Readln(tNaamLeer, sReel); arrDoolhof[indeks] :=
  sReel; redV4.Lines.Add(IntToStr(iIndeks) + #9 +
  sReel); Inc(iIndeks);
end;
end;
end;

```

---

**Vraag 4.2 = 19 punte**

---

```

procedure TfrmV4.btnV4_2Click(Sender: TObject);
var
  iRy, iKol, iTel, iLangsteInReel, iMaks: integer;
  arrRye: array [1..iRyTel] of integer;
  sRy: String;
begin
  iMaks := 0;
  for iRy := 1 to iRyTel do
  begin
    sRy := arrDoolhof[iRy];

    iLangsteInReel := 0;
    iTel := 0;
    for iKol := 1 to Length(sRy) do
    begin
      if sRy[iKol] = '-' then
      begin
        Inc(iTel);
        if iTel > iLangsteInReel then
          iLangsteInReel := iTel
        end
      else
        iTel := 0;
      end;
    end;
    arrRye[iRy] := iLangsteInReel;

    if iMaks < arrRye[iRy] then
      iMaks := arrRye[iRy];
    end;

    redV4.Lines.Add('');
    redV4.Lines.Add('Langste gang(e) met ' + IntToStr(iMaks)
    + ' spasies in ry(e):');
    for iRy := 1 to Length(arrRye) do
    begin
      if arrRye[iRy] = iMaks then
        redV4.Lines.Add(' ' + IntToStr(iRy));
      end;
    end;
  end;
end.

```