

SA's Leading Past Year

Exam Paper Portal



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



**SA EXAM
PAPERS**



Province of the
EASTERN CAPE
EDUCATION



NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

SEPTEMBER 2022

**INLIGTINGSTEGNOLOGIE V1
NASIENRIGLYN**

PUNTE: 150

Hierdie nasienriglyn bestaan uit 16 bladsye.

NAAM VAN LEERDER:				
TOTAAL VRAAG 1:	TOTAAL VRAAG 2:	TOTAAL VRAAG 3:	TOTAAL VRAAG 4:	TOTAAL
/40	/40	/40	/30	/150

VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE		MAKS. PUNTE	PUNTE BEHAAL
1.1	Knoppie [1.1 Vorm] shpToets.Shape := stcircle; ✓ shpToets.Brush.Color := clgreen; ✓ gbxVraag12.Visible := true; ✓ gbxVraag13.Enabled := true; ✓	4	
1.2	Knoppie [1.2 Ontsyfer] Kry toevoer van edtToevoer ✓ Inisialiseer 'n leë string ✓ Lus totdat toevoer string leeg is ✓ Kry ordinale getal en verwerk na integer ✓ Verwyder ordinale getal van toevoer string ✓ Verwerk ordinale getal na karakter deur CHR te gebruik ✓ Voeg karakter by nuwe string ✓ Vertoon nuwe string in edtAfvoer ✓	8	
1.3	Knoppie [1.3 Toets] Maak die list box skoon ✓ Kry toevoer van spinedit ✓ Inisialiseer die teller vir skikking ✓ <i>Kry die aantal priemfaktore van die toevoer en stoor in skikking – 7 punte:</i> Buitenste Lus van 1 tot toevoer ✓ Stel getal priemfaktore na 0 ✓ Binneste Lus van 1 na buitenste lus telle (geneste lus) ✓ As binneste lus teler 'n priemfaktor is ✓ Vermeerder dan die aantal priemfaktore met 1 ✓ As getal priemfaktore = 2 ✓ Voeg dan die buitenste lus teller by skikking ✓		

	Bereken en vertoon die pare van priemfaktore wat se som gelyk is aan die toevoer – 9 punte: Stel boolse vlag na vals ✓ Buitenste lus van 1 na skikking teller – 1 ✓ Stel veranderlike 1 na skikking[buitenste lus teller] ✓ Binneste lus van buitenste loop +1 na skikking teller ✓ Stel veranderlike 2 na skikking[binneste lus teller] ✓ As veranderlike 1 + veranderlike 2 = toevoer ✓ Vertoon dan albei getalle in listbox ✓ Omgeskakel na string ✓ Stel Boolse vlag na waar ✓ As Boolse vlag = vals is ✓ Vertoon gepaste boodskap in ShowMessage komponent ✓	21	
1.4	Knoppie [1.4 Area] Bereken area van vierkant ✓ Bereken area van sirkel: $\text{Pi} * (0.9/(2 * \text{Pi})) * (0.9/(2 * \text{Pi}))$ ✓ gebruik Pi of 3.1415 ✓ Trek area van vierkant af van area van sirkel ✓ Vermenigvuldig met 100 ✓ Voeg antwoord by huidige 'caption' van lblAfvoer ✓ Rond af na 3 desimale plekke ✓	7	
TOTAAL VRAAG 1		40	

VRAAG 2: DATABASISPROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
2.1.1	Knoppie: [V 2.1.1]	3	
	'select Afleveringsadres, AfleveringDatum from tblBestellings order by AfleveringDatum Desc'		
	Konseppte: SELECT twee korrekte velde ✓ FROM korrekte tabel ✓ ORDER BY korrekte veld DESC ✓		
2.1.2	Knoppie: [V 2.1.2]	4	
	'select count(BestellingID) as Items20ofMeer from tblBestellings where items >= 20'		
	Konseppte: SELECT count(enige veld van tblBestellings-tabel) ✓ as Items20ofMeer ✓ FROM korrekte tabel ✓ WHERE items >=20 ✓		
2.1.3	Knoppie: [V 2.1.3]	4	
	'select * from TblBestellings where Afleveringsadres like ' + quotedstr('%' + sline + '%')		
	Konseppte: SELECT * (alle velde) ✓ FROM korrekte tabel ✓ WHERE Afleveringsadres LIKE ✓ quotedstr('%' + sline + '%') ✓		
2.1.4	Knoppie: [V 2.1.4]	3	
	'update tblBestellings set Kollekteer = true where AfleveringDatum is null'		
	Konseppte: Update tblBestellings ✓ Set Kollekteer = true ✓ Where AfleveringDatum is null ✓		
2.1.5	Knoppie: [V 2.1.5]	4	
	'select Afleveringsadres, AfleveringDatum - BestellingDatum as AflewerDae from TblBestellings where Kollekteer = false'		
	Konseppte: Select DeliveryAdress ✓ AfleveringDatum – BestellingDatum ✓ From TblBestellings ✓ Where Kollekteer = false ✓		

2.1.6	Knoppie: [V 2.1.6] 'select WinkelNaam, format(sum(BestellingBedrag,"currency")) as TOTALEVerkope from tblBestellings, tblWinkels where tblWinkels.Winkelsid = tblBestellings.Winkelsid group by WinkelNaam	7	
	Concepts: Select WinkelNaam, ✓ format(sum(BestellingBedrag ✓, "currency")) ✓ as TOTALEVerkope ✓ from TblBestellings, Winkels ✓ where TblWinkels.TblWinkelsID = TblBestellings.WinkelsID ✓ group by WinkelNaam ✓		
	2.1 Subtotaal: SQL	25	
2.2.1	Knoppie: [V 2.2.1] tblWinkels.First ✓ while not tblWinkels.eof do ✓ if tblWinkels['Aanlyn'] = true then ✓ redOutput.Lines.Add(tblWinkels['WinkelNaam']) ✓ tblWinkels.Next ✓	5	
2.2.2	Knoppie: [V 2.2.2] Gaan na eerste ry in tblWinkels en lus ✓ as WinkelNaam = combobox item ✓ stel dan integer veranderlike na tblWinkelsID ✓ Gaan na volgende ry in tblWinkels Stel reële som veranderlike na = 0 ✓ Gaan na eerste ry in tblBestellings en lus as TblWinkelsID = integer veranderlike van eerste lus ✓ Vertoon dan in redAfvoer: BestellingDatum omgeskakel na string ✓ en BestellingBedrag in geldeenheid ✓ Voeg BestellingBedrag by reële som veranderlike ✓ Gaan na volgende ry in TblBestellings ✓ Na die lus vertoon die reële som veranderlike in redAfvoer in geldeenheid ✓	10	
	2.2 Subtotaal: Kode Konstruke	15	
	TOTAAL VRAAG 2	40	

VRAAG 3: OBJEK-GEÏORIENTEERDE PROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
3.1.1	Konstruktor Create: Korrekte naam ✓ met een string parameter en een integer parameter ✓ Ken korrekte parameter waardes toe aan korrekte attribute ✓ Stel fKoste = 70 * fUre ✓	4	
3.1.2	prosedure setTotaleKoste Korrekte metode – procedure ✓ Een integer parameter ✓ Stel korrekte attribuut (fTotaleKoste) ✓ = 70 * parameter ✓	4	
3.1.3	Funksie BerekenDae(ure : integer) : integer Korrekte metode – integer funksie ✓ een integer parameter ✓ As ure > 8 ✓ bereken en stuur korrekte dae terug (Ceil of ander metode) ✓ else ✓ Stuur 1 dag terug ✓	6	
3.1.4	funksie toString : string; korrekte string metode ✓ korrekte attribute teruggestuur as result ✓ gebruik #13, ✓ skakel ure en koste om na string ✓	4	
	2.1 Subtotaal: Objekklas	18	
3.2.1	Knoppie [V 3.2.1 Voeg by kwotasie] Kry toevoere van spinedit ✓ en list box ✓ Instansieer die objek Objeknaam = ✓ tHerstel.create ✓ een string, een integer parameter ✓ In korrekte volgorde ✓ Gebruik metode van klas ✓ en objeknaam om te vertoon ✓ Vertoon leë reël in redV3 ✓ Gebruik metode van klas en objeknaam ✓ om by totale ure globale veranderlike te voeg ✓	11	
3.2.2	Knoppie [V 3.2.2 Finaliseer Kwotasie] Gebruik objeknaam en metode van die klas ✓ setTotaleKoste met globale totale ure as parameter ✓ Gebruik objeknaam en metode van die klas ✓ getTotaleKoste en ken aan lokale veranderlike toe ✓ As cbxParte geselekteer is, ✓ voeg 150 by totale koste ✓ Vertoon dae in richedit deur objeknaam ✓ en korrekte metode BerekenDae met iTotaleUre as parameter ✓ as 'n string ✓ Vertoon totale koste in richedit ✓ omgeskakel na geldeenheid ✓	11	
	2.2 Subtotaal: Vormklas	22	
	TOTAAL VRAAG 3	40	

VRAAG 4: PROBLEEMOPLOSSING		MAKS. PUNTE	PUNTE BEHAAL
4.1	Assignfile stelling ✓ Reset teksleër vir lees ✓ Stel kolomteller na 0 ✓ Lus deur teksleër ✓ Lees 'n reël ✓ Inkrementeer kolom ✓ Stel ry = 1 ✓ Voeg reël na skikking, ✓ verander na integer ✓ Lees 'n reël ✓ Stel ry = 2 ✓ Voeg reël na skikking, verander na integer ✓	12	
4.2	Stel tellers van A en B na 0 ✓ Stel kolomteller na 0 ✓ Lus deur skikking ✓ Stel ry = 1 ✓ Voeg 1 by kolom teller ✓ Stel eerste syfer na skikking gebruik ry en kolom tellers [ry,kolom] ✓ Stel ry = 2 ✓ Stel tweede syfer na skikking gebruik ry en kolom-tellers [ry,kolom] ✓ as eerste syfer > tweede syfer ✓ vertoon in richedit korrekte bewoording – distriknommer en A ✓ inkrementeer A teller ✓ else ✓ vertoon in richedit korrekte bewoording – distriknommer en B ✓ inkrementeer B teller ✓ as A teller > B teller ✓ dan vertoon wenners A ✓ else ✓ vertoon wenners B ✓	18	
TOTAAL VRAAG 4		30	

VOORBEELDE VAN OPLOSSINGS**VRAAG 1**

```
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var inum, i1,i2, k, x, m, ifactors : integer;
arrprime : array[1..100] of integer;
icount: integer;
bprime : boolean;
begin
  lstoutput.clear;

  icount := 0;
  inum := sednumber.Value;

  for k := 1 to inum do
    begin
      ifactors := 0;
      for m := 1 to k do
        begin
          if k mod m = 0 then
            inc(ifactors);
          end;
        if ifactors = 2 then
          begin
            inc(icount);
            arrprime[icount] := k;
          end;
        end;
      end;

  bprime := false;
  for k := 1 to icount - 1 do
    begin
      i1 := arrprime[k];
      for x := k + 1 to icount do
        begin
          i2 := arrprime[x];
          if i1 + i2 = inum then
            begin
              lstoutput.Items.Add(inttostr(i1) + ' + ' + inttostr(i2));
              bprime := true;
            end;
          end;
        end;
      end;

  if bprime = false then
    showmessage(inttostr(inum) + ' cannot be calculated using the sum of two prime
    numbers')
  end;
```

```
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var rsq, rc : real;
rr : real;
begin
rsq := 0.2 * 0.2;
rc := Pi * (0.9/(2 * Pi)) * (0.9/(2 * Pi));
rr := (rc - rsq) * 100;
lbloutput.Caption := lbloutput.Caption + floattostrf(rr,ffixed,8,3) + ' square cm';
end;
```

```
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
begin
shpToets.Shape := stcircle;
shpToets.Brush.Color := clgreen;
gbquestion1_2.Visible := true;
gbquestion1_3.Enabled := true;
end;
```

```
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var sline : string;
snew , sword : string;
k, inum : integer;
begin
// If text in edtinput is erased use the following as input:
//      '73 84 32 105 115 32 101 97 115 121 33'
```

```
sline := edtinput.Text;
snew := "";
while length(sline) <> 0 do
begin
    sword := copy(sline,1, pos(' ',sline) - 1);
    delete(sline,1, pos(' ',sline));
    inum := strtoint(sword);
    snew := snew + chr(inum);
end;
edtoutput.text := snew;
end;

end.
```

VRAAG 2

```
procedure TQuestion_2.btnQuestion2_1_1Click(Sender: TObject);
// Provided code - do not change
var
    sSQL1: String;
begin
    /// enter your code below//
    sSQL1 := 'select Afleveringsadres, AfleveringDatum from tblBestellings order by
AfleveringDatum desc';

    // Provided code - do not change
    dbCONN.runSQL(sSQL1);
    if length(ssql1) <> 0 then
        dbconn.setgridforsql1(dbgsq1);
end;

procedure TQuestion_2.btnQuestion2_1_2Click(Sender: TObject);
// Provided code - do not change
var
    sSQL2: String;
begin
    /// enter your code below//
    sSQL2 := 'select count(BestellingID) as TblBestellings20orMore from tblBestellings
where items >= 20';

    // Provided code - do not change
    dbCONN.runSQL(sSQL2);
end;

procedure TQuestion_2.btnQuestion2_1_3Click(Sender: TObject);
// Provided code - do not change
var
    sline : string;
    sSQL3: String;
begin
    sline := inputbox('Enter a street name','Nxolo Street');
    /// enter your code below//
    sSQL3 := 'select * from TblBestellings where Afleveringsadres like ' + quotedstr('%' +
sline + '%)';

    // Provided code - do not change
    dbCONN.runSQL(sSQL3);
    if length(ssql3) <> 0 then
        dbconn.setgridforsql3(dbgsq1);
end;

procedure TQuestion_2.btnQuestion2_1_4Click(Sender: TObject);
// Provided code - do not change
var
    sSQL4: String;
begin
    /// enter your code below//
    sSQL4 := 'update tblBestellings set Kollekteer = true where AfleveringDatum is null';
```

```
// Provided code - do not change
dbCONN.executeSQL(sSQL4,dbgtblWinkels,dbgorder,dbgsql);
if length(ssql4) <> 0 then
  dbconn.setgridforsql3(dbgsql);
end;

procedure TQuestion_2.btnQuestion2_1_5Click(Sender: TObject);
// Provided code - do not change
var
  sSQL5: String;
begin
  /// enter your code below//
  sSQL5 := 'select Afleveringsadres, AfleveringDatum - BestellingDatum as AfleverDae
from TblBestellings where Kollekteer = false';

  // Provided code - do not change
  dbCONN.runSQL(sSQL5);
  if length(ssql5) <> 0 then
    dbconn.setgridforsql5(dbgsql);
  end;

procedure TQuestion_2.btnQuestion2_1_6Click(Sender: TObject);
// Provided code - do not change
var
  sSQL6: String;
begin
  /// enter your code below//
  sSQL6 := 'select WinkelNaam, format(sum(BestellingBedrag),"currency") as
TOTALEVerkope from tblBestellings, tblWinkels where tblWinkels.tblWinkelsid =
tblBestellings.tblWinkelsid group by WinkelNaam';

  // Provided code - do not change
  dbCONN.runSQL(sSQL6);
  if length(ssql6) <> 0 then
    dbconn.setgridforsql6(dbgsql);
  end;

procedure TQuestion_2.btnQuestion2_2_1Click(Sender: TObject);
begin
  // Provided code - do not change
  redoutput.Clear;
  redoutput.Lines.Add('TblWinkelss that are available for Aanlyn tblWinkelsping');
  redoutput.Lines.Add('-----');
  /// enter your code below//
  tblWinkels.First;
  while not tblWinkels.eof do
    begin
      if tblWinkels['Aanlyn'] = true then
        redoutput.Lines.Add(tblWinkels['WinkelNaam']);
      tblWinkels.Next;
    end;
  end;
end;
```

```
procedure TQuestion_2.btnQuestion2_2_2Click(Sender: TObject);
var stblWinkels : string; // Provided code - do not change
bfound : boolean;
inum : integer;
rsum : real;
begin
// Provided code - do not change
redoutput.Clear;
redoutput.Lines.Add('Order Date' + #9 + 'Amount of order');
redoutput.Lines.Add('-----') ;
stblWinkels := cmbtblWinkels.Text;
/// enter your code below//
bfound := false;
tblWinkels.First;
while (not tblWinkels.Eof) and (bfound = false) do
begin
if tblWinkels['WinkelNaam'] = stblWinkels then
begin
inum := tblWinkels['tblWinkelsid'];
bfound := true;
end;
tblWinkels.Next;
end;
rsum := 0;
tblBestellings.first;
while not tblBestellings.eof do
begin
if tblBestellings['tblWinkelsid'] = inum then
begin
redoutput.Lines.Add(datetostr(tblBestellings['BestellingDatum']) + #9 +
floattostf(tblBestellings['BestellingBedrag'],ffcurrency,8,2));
rsum := rsum + tblBestellings['BestellingBedrag'];
end;
tblBestellings.Next;
end;
redoutput.Lines.Add("");
redoutput.Lines.Add('Total amount ordered: ' + floattostf(rsum,ffcurrency,8,2));
end;
```

VRAAG 3**Klaseenheid:**

unit Question3ClassDefinition;

interface

uses sysutils, math, dialogs;

type

TRepair = class

private

frepairname : string;

fhours : integer;

fcost : real;

ftotalcost : real;

public

constructor create(sname : string; ihrs : integer);

function calculatedays(ihrs : integer) : integer;

function gethours : integer;

function tostring : string;

procedure settotalcost(ihrs : integer);

function gettotalcost : real;

end;

implementation

{ TRepair }

constructor TRepair.create(sname: string; ihrs: integer);

begin

frepairname := sname;

fhours := ihrs;

fcost := 70 * fhours;

end;

// provided code do not delete//

function TRepair.gethours: integer;

begin

result := fhours;

end;

// provided code do not delete//

function TRepair.gettotalcost: real;

begin

result := ftotalcost;

end;

procedure TRepair.settotalcost(ihrs: integer);

begin

ftotalcost := ihrs * 70;

end;

```
function TRepair.calculatedays(ihrs: integer) : integer;
var idays : integer;
begin
  if ihrs > 8 then
    begin
      idays := ceil(ihrs/8);
    end
  else
    idays := 1;
  result := idays;
end;
```

```
function TRepair.tostring: string;
begin
  result := frepairname +
    #13 + 'Hours to complete: ' + inttostr(fhours) +
    #13 + 'Labour Cost: ' + floattostrf(fcost,ffcurrency,8,2)
end;

end.
```

Hoofteenheid:

```
procedure TForm1.btnQ3_2_1Click(Sender: TObject);
begin
  objrepair := trepair.create(lstrepairs.Items[lstrepairs.ItemIndex],sedhours.value);
  redq3.Lines.Add(objrepair.tostring);
  redq3.Lines.Add("");
  itotalhours := itotalhours + objrepair.gethours;
end;
```

```
procedure TForm1.cbxBuyPartsClick(Sender: TObject);
var rcost : real;
begin
  objrepair.settotalcost(itotalhours);
  rcost := objrepair.gettotalcost;
```

```
if cbxparts.checked = true then
  rcost := rcost + 150;
```

```
redq3.Lines.Add("");
redq3.Lines.Add('Total days to complete all jobs = ' +
  inttostr(objrepair.calculatedays(itotalhours)));
redq3.Lines.Add('Total labour cost = ' + floattostrf(rcost,ffcurrency,8,2));
end;
```

VRAAG 4

```
unit Question4_u;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, ComCtrls;

type
  TForm1 = class(TForm)
    pgc: TPageControl;
    tbsvoting: TTabSheet;
    btnQ4_1: TButton;
    btnQ4_2: TButton;
    redQ4: TRichEdit;
    procedure btnQ4_1Click(Sender: TObject);
    procedure btnQ4_2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  ar2Stemme : array[1..2,1..10] of integer;

implementation
{$R *.dfm}

procedure TForm1.btnQ4_1Click(Sender: TObject);
var myfile : textfile;
    irow, icol, inum: integer;
    sline : string;
begin
  assignfile(myfile, 'stemme.txt');
  reset(myfile);
  icol := 0;
  while not eof(myfile) do
    begin
      readln(myfile, sline);
      inc(icol);
      irow := 1;
      ar2Stemme[irow, icol] := strtoint(sline);
      readln(myfile, sline);
      irow := 2;
      ar2Stemme[irow, icol] := strtoint(sline);
    end;
  end;
end;
```



```
procedure TForm1.btnQ4_2Click(Sender: TObject);
var
irow, icol, inum1, inum2, k, ia, ib : integer;
begin
    icol := 0;
    ia := 0;
    ib := 0;
    while icol < 10 do
        begin
            irow := 1 ;
            inc(icol);
            inum1 := ar2Stemme[irow,icol] ;
            irow := 2;
            inum2 := ar2Stemme[irow,icol];
            if inum1 > inum2 then
                begin
                    redq4.Lines.Add('Distrik '+ inttostr(icol) + ' wenner is A');
                    inc(ia);
                end
            else
                begin
                    redq4.Lines.Add('Distrik '+ inttostr(icol) + ' wenner is B');
                    inc(ib);
                end;
            end;
            redq4.Lines.Add("");
        if ia > ib then
            redq4.Lines.Add('Die algehele wenner is A')
        else
            redq4.Lines.Add('Die algehele wenner is B')
        end;
    end;

end.
```