



Province of the  
**EASTERN CAPE**  
EDUCATION

**NASIONALE  
SENIOR SERTIFIKAAT**

**GRAAD 11**

**NOVEMBER 2018**

**INLIGTINGSTEGNOLOGIE V1  
NASIENRIGLYN**

**PUNTE: 150**

---

Hierdie nasienriglyn bestaan uit 17 bladsye.

---

<b>NAAM VAN LEERDER:</b>				
<b>TOTAAL VRAAG 1</b>	<b>TOTAAL VRAAG 2</b>	<b>TOTAAL VRAAG 3</b>	<b>TOTAAL VRAAG 4</b>	<b>TOTAAL</b>
/44	/22	/30	/54	/150

<b>VRAAG 1</b>		<b>MAKS PUNTE</b>	<b>PUNTE BEHAAL</b>
1.1	<b>KNOPPIE: [VRAAG 1.1]</b>  Kry die kelner se naam en van ✓ Gebruik die pos funksie ✓ om die van te kry ✓ Skep die verwysing: eerste letter van naam ✓ en '.' ✓ en van ✓ Kry ewekansige kode ✓ korrekte omvang ✓ Vertoon die verwysing ✓ Vertoon die kode ✓	10	
1.2	<b>KNOPPIE: [VRAAG 1.2]</b>  Kry die ID nommer ✓ Kry die lengte van die ID nommer ✓ As die lengte gelyk is aan 13 ✓ Lus deur die ID ✓ Toets of karakters syfers is of nie ✓ As geen letter in ID ✓ Kry stelseldatum ✓ Bereken die ouderdom: Bepaal die JJJJ deur ID te gebruik (1900 of 2000) ✓✓ Kry die JJJJ van stelseldatum ✓ Trek af van mekaar ✓ If/case stelling ✓ Korrekte omvang ✓ Vertoon korrekte boodskap ✓ As ID nie korrek is nie ✓ Vertoon 'n boodskap ✓	16	
1.3	<b>KNOPPIE: [VRAAG 1.3]</b>  Kry die ure ✓ As die checkbox gekies is ✓ Kry die oortydure van inputbox ✓ as 'n getal ✓ Bereken normale ure se salaris ✓ Bereken oortydure se salaris ✓ Tel bymekaar ✓ Vertoon totale salaris ✓ geformateer na geldeenheid en twee desimale plekke ✓ Verander die paneel se kleur na skyblue ✓	10	

1.4	<b>KNOPPIE: [VRAAG 1.4]</b>  Kry die item van combobox✓ If/Case stelling✓ Kry die grootte✓✓ Vertoon die bestelling opskrif ✓ item✓ en grootte in hakies ✓ en Hoed oor drie reëls✓	<b>8</b>	
		<b>44</b>	

VRAAG 2		MAKS PUNTE	PUNTE BEHAAL
2.1	tblTafels.first; ✓ while not tblTafels.eof do ✓ begin rtotaal := rtotaal ✓ + tblTafels['TafelBedragBetaal']; ✓ tblTafels.next; ✓ end; Vertoon totale bedrag ✓	6	
2.2	tblTafels.Sort := 'KelnerID'; ✓	1	
2.3	Kry die kelnerID van inputbox ✓ tblTafels.First; ✓ Inisialiseer die totaal veranderlike ✓ Vertoon die KelnerID as 'n opskrif ✓ Lus ✓ if skelner = tblTafels['KelnerID'] then ✓ Vertoon die TafelID ✓ Inkrementeer die totaal veranderlike ✓ Gaan na volgende rekord ✓ Vertoon die totale aantal tafels ✓	10	
2.4	tblTafels.Insert; ✓ tblTafels['TafelID'] := 71; ✓ tblTafels['KelnerID'] := 10; ✓ tblTafels['TafelGaste'] := 5; ✓ tblTafels.Post; ✓	5	
		22	

VRAAG 3		MAKS PUNTE	PUNTE BEHAAL
3.1	Assignfile vir ShoppingList.txt✓ Rewrite✓  As die lêer (Stocklist.txt) nie bestaan nie✓ Vertoon 'n boodskap en exit✓ Assignfile vir Stocklist.txt✓ Reset✓  Vertoon 'n opskrif✓  Lus✓ Lees van lêer ✓ Vind die posisie van #✓ Copy die item ✓ Delete die item✓ Kry die huidige bedrag as 'n getal (integer) ✓ Kry die eenheid✓ Kry die minimum ✓ Kry die maksimum✓ As die huidige bedrag minder is as minimum ✓ Bereken wat benodig word✓ Rond die bedrag wat benodig word na die naaste Heelgetal ✓ (rbenodig := ceil(rbenodig)); As eenheid kg✓ of bottels is✓ Vertoon die bedrag benodig ✓, eenheid✓ en item✓ Skryf na lêer✓ Else✓ Vertoon bedrag benodig ✓ en item✓ Skryf na lêer✓ Vertoon boodskap dat inkopielys gestoor is ✓	30	
		30	

VRAAG 4		MAKS PUNTE	PUNTE BEHAAL
4.1	Voeg prosedure onder public✓ Prosedure Sorteër opskrif✓ Outer loop✓ Inner Loop✓ If arritem[k] > arritem[l] ✓ Swop arritem ✓✓✓ Swop arrkoste ✓✓✓	11	
4.2	Inisialiseer kos teller✓ Inisialiseer drinkgoed teller✓ Roep die Sorteër metode✓ Lus ✓ Kry die posisie van die #✓ As dit 'n kos item is(K) ✓ Inkrementeer die kos teller ✓ Ken item aan skikking toe vir later in program ✓ Ken koste aan skikking toe ✓ As dit 'n Drinkgoed item is(D) ✓ Inkrementeer drinkgoed teller✓ Ken item aan skikking toe vir later in program ✓ Ken koste aan skikking toe ✓  Vertoon opskrif KOS✓ Lus✓ korrekte aantal kere ✓ Vertoon kos items✓ Vertoon opskrif DRINKGOED✓ Lus✓ korrekte aantal kere Vertoon drinkgoed items✓	20	
4.3	Maak richedit skoon✓ Stel tabcount na 1✓ Stel tab na 150✓ Vertoon opskrif ✓ Lus✓ Bereken BTW✓ en wins (markup)✓ Rond op (ceil) ✓ Vertoon die prys✓ geformateer na geldeenheid ✓	10	
4.4.1	function Gevind(sitem : string✓) : boolean; ✓ Stel Boolse veranderlike na false✓ Lus✓ As item gevind is ✓ Verander Boolse veranderlike na true✓ Ken Boolse veranderlike toe aan funksie/result ✓	7	
4.4.2	Kry item van inputbox✓ Roep die Gevind metode ✓ in if stelling✓ Vertoon boodskap (gevind) ✓ Else✓ Vertoon boodskap (nie gevind) ✓	6	
		54	

**MOONTLIKE OPLOSSINGS****VRAAG 1**

unit Question1\_u;

interface

uses

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

type

```
TfrmQuestion1 = class(TForm)
  gbxQ1_1: TGroupBox;
  btnQ1_1: TButton;
  edtWaiter: TLabeledEdit;
  edtRef: TLabeledEdit;
  edtCode: TLabeledEdit;
  gbxQ1_2: TGroupBox;
  edtID: TLabeledEdit;
  btnQ1_2: TButton;
  redQ1_2: TRichEdit;
  gbxQ1_3: TGroupBox;
  gbxQ1_4: TGroupBox;
  sedHours: TSpinEdit;
  lblHours: TLabel;
  cbxOvertime: TCheckBox;
  btnQ1_3: TButton;
  pnlOutput: TPanel;
  cmbItem: TComboBox;
  btnQ1_4: TButton;
  grpSizes: TRadioGroup;
  procedure btnQ1_1Click(Sender: TObject);
  procedure btnQ1_2Click(Sender: TObject);
  procedure btnQ1_3Click(Sender: TObject);
  procedure btnQ1_4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

frmQuestion1: TfrmQuestion1;

implementation

{\$R \*.dfm}

procedure TfrmQuestion1.btnQ1\_1Click(Sender: TObject);

var

```
swaiter, sname, ssurname, sref : string;
ipos, icode : integer;
begin
  swaiter := edtwaiter.text;
  ipos := pos(' ',swaiter);
  sname := copy(swaiter, 1, ipos - 1);
  delete(swaiter, 1, ipos);
  ssurname := swaiter;
  sref := sname[1]+'.'+ssurname;
  icode := randomrange(10000,100000);
  edtref.Text := sref;
  edtcode.text := inttostr(icode);
end;
```

```
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
  sid, sdate : string;
  k, iage: Integer;
  bletter : boolean;
begin
  bletter := false;
  sid := edtid.text;
  if length(sid) = 13 then
    begin
      for k := 1 to 13 do
        begin
          if not (strtoint(sid[k]) in [0..9]) then
            bletter := true;
          end;
        end;
      if bletter = false then
        begin
          sdate := datetostr(date());
          if sid[1] = '0' then
            iage := strtoint(copy(sdate,1,4)) - (2000+(strtoint(copy(sid,1,2))))
          else
            iage := strtoint(copy(sdate,1,4)) - (1900+(strtoint(copy(sid,1,2)))));
          case iage of
            0..15 : redQ1_2.Lines.Add('Too young to work');
            16..18 : redQ1_2.Lines.Add('Weekday x 1'+#13+'Weekend');
          else
            redQ1_2.lines.Add('Weekdays x 5'+#13+'Saturday OR Sunday');
          end;
        end;
      end
    else
      redQ1_2.Lines.Add('Incorrect ID');
    end;
end;
```

```
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
  ihours, iover : integer;
  rwage : real;
```



```
begin
  ihours := sedHours.Value;
  if cbxOvertime.checked then
    iover := strtoint(Inputbox(",",""))
  else
    iover := 0;
  rwage := ihours * 18.25 + 1.5*iover*18.25;
  pnlOutput.Caption := floattostrf(rwage,ffcurrency,10,2);
  pnlOutput.Color := clSkyBlue;
end;

procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var
  sitem, ssize : string;
begin
  sitem := cmbItem.text;
  case rgpSizes.itemindex of
    0 : ssize := 'XS';
    1 : ssize := 'S';
    2 : ssize := 'M';
    3 : ssize := 'L';
    4 : ssize := 'XL';
  end;
  Showmessage('Order:'+#13+sitem +'('+ssize+')'+#13+'Peak Cap');
end;

end.
```

**VRAAG 2**

```
unit Question2_u;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, Buttons, StdCtrls, ComCtrls, Grids, DBGrids, DB, ADODB;
```

```
type
```

```
  TForm1 = class(TForm)  
    tblTables: TADOTable;  
    DataSource1: TDataSource;  
    DBGrid1: TDBGrid;  
    Button1: TButton;  
    Button2: TButton;  
    Button3: TButton;  
    redoutput: TRichEdit;  
    Button4: TButton;  
    BitBtn1: TBitBtn;  
    procedure Button1Click(Sender: TObject);  
    procedure Button2Click(Sender: TObject);  
    procedure Button3Click(Sender: TObject);  
    procedure Button4Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;
```

```
var
```

```
  Form1: TForm1;
```

```
implementation
```

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

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  rtotal : real;
```

```
begin
```

```
  tblTables.open;
```

```
  tblTables.first;
```

```
  while not tblTables.eof do
```

```
  begin
```

```
    rtotal := rtotal + tblTables['TableAmountPaid'];
```

```
    tblTables.next;
```

```
  end;
```

```
  redoutput.Lines.Add('Total Amount: '+floattostrf(rtotal,ffcurrency,10,2));
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    tblTables.Sort := 'waiterid';
end;

procedure TForm1.Button3Click(Sender: TObject);
var
    swaiter : string;
    itables : integer;
begin
    swaiter := inputbox("", "");
    tblTables.Open;
    tblTables.First;
    itables := 0;
    Redoutput.lines.add('WaiterID: '+swaiter+#13);
    while not tblTables.eof do
    begin
        if swaiter = tblTables['waiterID'] then
        begin
            redoutput.lines.add(tblTables['TableID']);
            itables := itables + 1;
        end;
        tblTables.Next;
    end;
    redoutput.lines.add(#13+ 'Total Number of Tables: '+inttostr(itables));
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    tblTables.Insert;
    tblTables['TableID'] := 26;
    tblTables['WaiterID'] := 10;
    tblTables['TableGuests'] := 5;
    tblTables.Post;
end;

end.
```

**VRAAG 3**

```
unit Question3_u;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, ComCtrls, math;
```

```
type
```

```
  TfrmQuestion3 = class(TForm)  
    redoutput: TRichEdit;  
    btnList: TButton;  
    procedure btnListClick(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;
```

```
var
```

```
  frmQuestion3: TfrmQuestion3;
```

```
implementation
```

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

```
procedure TfrmQuestion3.btnListClick(Sender: TObject);
```

```
var
```

```
  icount, ipos : integer;  
  myfile, myfile2 : textfile;  
  rmin, rmax, rneed, rcurrent : real;  
  sitem, sunit, soneline : string;
```

```
begin
```

```
  Assignfile(myfile2, 'ShoppingList.txt');  
  Rewrite(myfile2);
```

```
  if fileexists('stocklist.txt') <> true then
```

```
  begin
```

```
    Showmessage('File does not exist');  
    Exit;
```

```
  end;
```

```
  Assignfile(myfile, 'stocklist.txt');
```

```
  Reset(myfile);
```

```
  redoutput.lines.Add('SHOPPING LIST'+#13);
```

```
  while not eof(myfile) do
```

```
  begin
```

```
    readln(myfile, soneline);  
    ipos := pos('#', soneline);  
    sitem := copy(soneline, 1, ipos-1);  
    delete(soneline, 1, ipos);
```

```
ipos := pos('#',soneline);
rcurrent := strtofloat(copy(soneline,1,ipos-1));
delete(soneline,1,ipos);
ipos := pos('#',soneline);
sunit := copy(soneline,1,ipos-1);
delete(soneline,1,ipos);
ipos := pos('#',soneline);
rmin := strtofloat(copy(soneline,1,ipos-1));
delete(soneline,1,ipos);
rmax := strtofloat(soneline);
if rcurrent < rmin then
  begin
    rneed := rmax - rcurrent;
    if frac(rneed) > 0 then
      rneed := ceil(rneed);
    if (sunit = 'kg') or (sunit = 'bottles') then
      begin
        redoutput.lines.add(floattostr(rneed)+' ' + sunit + ' ' + sitem);
        writeln(myfile2,floattostr(rneed)+' '+sunit+' ' +sitem);
      end
    else
      begin
        redoutput.lines.add(floattostr(rneed)+' ' + sitem);
        writeln(myfile2,floattostr(rneed)+' ' + sitem);
      end;
    end;
  end;
end;
redoutput.Lines.Add(#13+'Shopping list saved to file');
Closefile(myfile);
Closefile(myfile2);

end;

end.
```

**VRAAG 4**

```
unit Question4_u;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Buttons, StdCtrls, ComCtrls, math;
```

```
type
```

```
TfrmQuestion4 = class(TForm)  
  btnItems: TButton;  
  redoutput: TRichEdit;  
  BitBtn1: TBitBtn;  
  btnFoodPrice: TButton;  
  btnSearch: TButton;  
  procedure btnItemsClick(Sender: TObject);  
  procedure btnFoodPriceClick(Sender: TObject);  
  procedure btnSearchClick(Sender: TObject);  
  procedure FormCreate(Sender: TObject);  
private  
  { Private declarations }  
public  
  procedure Sort;  
  { Public declarations }  
end;
```

```
var
```

```
frmQuestion4: TfrmQuestion4;  
arritem : array[1..22] of string;  
arrcost : array[1..22] of real = (23.20, 24.72, 29.36, 22.23, 19.96, 20.30, 25.21, 18.55, 5.30,  
4.53, 4.22, 4.72, 5.18, 3.95, 33.20, 37.35, 14.11, 17.19, 12.29, 9.53, 33.91, 3.95);  
arrfood : array[1..50] of string;  
arrdrink : array[1..50] of string;  
arrfcost : array[1..50] of real;  
arrdcost : array[1..50] of real;  
ifood, idrink : integer;
```

```
implementation
```

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

```
function Found(sitem : string) : boolean;
```

```
var
```

```
  bfound : boolean;  
  k: Integer;  
begin  
  bfound := false;  
  for k := 1 to 22 do  
  begin  
    if pos(uppercase(sitem), uppercase(arritem[k])) > 0 then  
      bfound := true;  
  end;  
end;
```

```
    result := bfound;
end;
procedure TfrmQuestion4.Sort;
var
    l,k: Integer;
    stemp : string;
    rtemp : real;
begin
    for k := 1 to 21 do
        for l := k+1 to 22 do
            if arritem[k] > arritem[l] then
                begin
                    stemp := arritem[k];
                    arritem[k] := arritem[l];
                    arritem[l] := stemp;
                    rtemp := arrcost[k];
                    arrcost[k] := arrcost[l];
                    arrcost[l] := rtemp;
                end;
            end;
        end;
    end;

procedure TfrmQuestion4.btnFoodPriceClick(Sender: TObject);
var
    k: Integer;
    rprice : real;
begin
    redoutput.Clear;
    redoutput.Paragraph.TabCount := 1;
    redoutput.Paragraph.Tab[0] := 150;
    redoutput.Lines.Add('FOOD ITEM'+#9+'PRICE');
    for k := 1 to ifood do
        begin
            rprice := ceil(arrfcost[k] * 1.60);
            redoutput.Lines.Add(arrfood[k] + #9 + floattostrf(rprice, fcurrency,10,0));
        end;
    end;

procedure TfrmQuestion4.btnItemsClick(Sender: TObject);
var
    k, ipos : integer;
begin
    Sort;
    for k := 1 to 22 do
        begin
            ipos := pos('#',arritem[k]);
            if arritem[k][ipos+1] = 'F' then
                begin
                    inc(ifood);
                    arrfood[ifood] := copy(arritem[k], 1, ipos-1);
                    arrfcost[ifood] := arrcost[k];
                end
            else
                end;
        end;
    end;
```

```

begin
  inc(idrink);
  arrdrink[idrink] := copy(arritem[k], 1, ipos-1);
  arrdcost[idrink] := arrcost[k];
end;
end;

redoutput.Lines.Add('FOOD ITEMS' + #13);
for k := 1 to ifood do
  redoutput.Lines.Add(arrfood[k]);

redoutput.Lines.Add(#13+'DRINK ITEMS'+#13);
for k := 1 to idrink do
  redoutput.Lines.Add(arrdrink[k]);
end;

procedure TfrmQuestion4.btnSearchClick(Sender: TObject);
var
  sitem : string;
begin
  sitem := inputbox("", "");
  if found(sitem) = true then
    Showmessage('Item is on the menu')
  else
    Showmessage('Item is not on the menu');
end;

procedure TfrmQuestion4.FormCreate(Sender: TObject);
begin
  ifood := 0;
  idrink := 0;

  {***GIVEN CODE***DO NOT CHANGE IT***}
  arrItem[1] := 'Hamburger and Chips#F';
  arrItem[2] := 'Cheese Burger and Chips#F';
  arrItem[3] := 'Bacon Burger and Chips#F';
  arrItem[4] := 'Veggie Burger and Chips#F';
  arrItem[5] := 'Hamburger#F';
  arrItem[6] := 'Cheese Burger#F';
  arrItem[7] := 'Bacon Burger#F';
  arrItem[8] := 'Veggie Burger#F';
  arrItem[9] := 'Chips - Large Plate#F';
  arrItem[10] := 'Soda#D';
  arrItem[11] := 'Fresh Juice - Orange#D';
  arrItem[12] := 'Fresh Juice - Apple#D';
  arrItem[13] := 'Homemade Lemonade#D';
  arrItem[14] := 'Water - Sparkling (500ml)#D';
  arrItem[15] := 'Bacon and Egg Burger with Chips#F';
  arrItem[16] := 'Giant Burger with Chips#F';
  arrItem[17] := 'Salad - Greek#F';
  arrItem[18] := 'Salad - Chicken#F';
  arrItem[19] := 'Salad - Potato#F';

```



```
arrItem[20] := 'Salad - Pasta#F';  
  
arrItem[21] := 'Giant Burger#F';  
  arrItem[22] := 'Water - Still (500ml)#D';  
end;  
  
end.
```