



NASIONALE SENIOR SERTIFIKAAT-EKSAMEN
NOVEMBER 2019

INLIGTINGSTEGNOLOGIE: VRAESTEL II

NASIENRIGLYNE

Tyd: 3 uur

120 punte

Hierdie nasienriglyne is opgestel vir gebruik deur eksaminators en hulp-eksaminators van wie verwag word om almal 'n standaardiseringsvergadering by te woon om te verseker dat die riglyne konsekwent vertolk en toegepas word by die nasien van kandidate se skrifte.

Die IEB sal geen bespreking of korrespondensie oor enige nasienriglyne voer nie. Ons erken dat daar verskillende standpunte oor sommige aangeleenthede van beklemtoning of detail in die riglyne kan wees. Ons erken ook dat daar sonder die voordeel van die bywoning van 'n standaardiseringsvergadering verskillende vertolkings van die toepassing van die nasienriglyne kan wees.

AFDELING A**VRAAG 1****Vraag 1.1 (4)**

```
SELECT *
FROM STUDENT
WHERE IsSenior = TRUE
ORDER BY StudentNaam
```

Vraag 1.2 (4)

```
SELECT ProjekNaam FROM PROJEK beide SELECT en FROM korrek
WHERE ProjekNaam
LIKE
'%tutor%' % moet voor en na
'*tutor*' verskyn vir ACCESS - * moet voor en na verskyn
```

Vraag 1.3 (3)

```
SELECT RegistrasieID,
       OpgehoopteUre/GetalBygewoon korrekte velde
       korrekte verdeling
FROM REGISTRASIE
```

Vraag 1.4 (4)

```
SELECT DISTINCT
SUBSTR (ProjekNaam,1,3) ACCESS: LEFT (ProjekNaam, 3)
FROM PROJEK
```

ALTERNATIEF:

```
SELECT SUBSTR (ProjekNaam,1,3) ACCESS: LEFT (ProjekNaam, 3)
FROM PROJEK
GROUP BY SUBSTR(ProjekNaam,1,3) ACCESS: LEFT(ProjekNaam, 3)
```

Vraag 1.5 (4)

```
SELECT ProjekNaam, StudentNaam Beide velde korrek
FROM STUDENT, PROJEK
WHERE StudentLeierID = StudentID
```

Vraag 1.6 (5)

```
SELECT StudentID, SUM (OpgehoopteUre)
       AS TOTAALURE
FROM REGISTRASIE
GROUP BY StudentID
```

Vraag 1.7 (4)

```
SELECT StudentNaam FROM STUDENT
WHERE StudentID
  NOT IN Beide NOT en IN
  (SELECT StudentID
   FROM REGISTRASIE
  )
```

ALTERNATIEF:

```
SELECT StudentNaam FROM STUDENT
LEFT JOIN REGISTRASIE
ON STUDENT.StudentID = REGISTRASIE.StudentID
WHERE RegistrasieID
  IS NULL
```

Vraag 1.8 (4)

```
SELECT ProjekNaam FROM PROJEK
WHERE DatumBegin =
( SELECT MAX (DatumBegin)
  FROM PROJEK
)
```

Vraag 1.9 (8)

```
SELECT StudentNaam, Count(*) AS GetalProjekte
FROM REGISTRASIE, STUDENT Beide tabelle
WHERE STUDENT.StudentID=REGISTRASIE.StudentID
GROUP BY StudentNaam
HAVING COUNT(*) >= 2
```

JAVA-OPLOSSING**VRAAG 2 STUDENT-KLAS****// Vraag 2.1 - 4**

```
public class Student // klasopskrif
{
  private String volleNaam; // alle kenmerke privaat
  private int graad; // almal korrek benoem
  private String belangstelling; // korrekte tipes
  private double ure;
```

// Vraag 2.2 - 3

```
public Student(String inFn, int inG, String inI, double inH)
// korrekte opskrif en parametername
{
  volleNaam = inFn; // velde tot parameters gestel
  graad = inG;
```

```

        belangstelling = inI;
        ure = inH;
    }

```

// Vraag 2.3 - 2

```

public int kryGraad() // korrekte opskrifte vir al drie metodes
{
    return graad;
}
public String kryBelangstelling() // korrekte terugsendings vir al drie
                                // metodes
{
    return belangstelling;
}
public double kryUre()
{
    return ure;
}

```

// Vraag 2.4 - 4

```

private String veranderNaam() // korrekte opskrif
{
    String temp = "";
    Scanner sc = new Scanner(volleNaam);
    String fName = sc.next(); // isoleer van
    String sName = sc.next(); // isoleer voor naam & voorletter
    return sName + ", " + fName.charAt(0);
                                // kombineer in die korrekte formaat
}

```

// Vraag 2.5 - 3

```

@Override
public String toString() // korrekte opskrif
{
    return veranderNaam() + "\t" + graad + "\t" + belangstelling + "\t" +
ure;
                                // roep veranderNaam()
                                // korrekte formaat met tabelkarakters
}

```

VRAAG 3 GRAAD-KLAS**// Vraag 3.1 - 2**

```

public class Graad // korrekte naam
{
    private int graad; // korrekte velde graad en totaal
    private double totaal;
}

```

// Vraag 3.2 - 2

```

private final int PERK = 20; // korrekte waardes toegeken
private final int BONUS = 10; // konstante velde verklaar

```

// Vraag 3.3 - 4

```

public Graad(int inG, double inT)
{
}

```

```

    graad = inG; // ken graad-veld toe
    if (inT > PERK) // vergelyk inT met PERK
        totaal = inT + BONUS; // tel BONUS by totaal wanneer > PERK
    else totaal = inT; // andersins ken inT toe aan totaal
}

```

// Vraag 3.4 - 2

```

public double kryTotaal() // korrekte toegangermetode
{
    return totaal;
}

public void stelTotaal(double inT) // korrekte mutatormetode
{
    totaal = inT;
}

```

// Vraag 3.5 - 2

```

@Override
public String toString() // korrekte opskrif
{
    return "Graad:" + graad + " totale ure " + totaal;
    // velde korrek gekombineer
}
}

```

VRAAG 4, 6 & 7 PROJEK-KLAS**// Vraag 4.1 - 2**

```

public class Projek // korrekte opskrif
{
    private String naam; // velde naam en maks korrek verklaar
    private int maks;
}

```

// Vraag 4.2 - 4

```

private Student sSkik[] = new Student[50]; // korrekte objeknaam
// 50 elemente verklaar
private int sTelling = 0; // sTelling korrek verklaar
private Graad gSkik[] = new Graad[5]; // gSkik korrek verklaar

```

// Vraag 4.3 - 8

```

public Projek(String inN, String inC, int inM)
{
    naam = inN; // velde naam en maks korrek toegeken
    maks = inM;
    try
    {
        Scanner scFile = new Scanner(new FileReader("Studente.txt"));
        // lêer oopgemaak om te lees

        String nm, i;
        int g;
        double h;
        while (scFile.hasNextLine()) // lus om lêer te lees

```

```

    {
        String line = scFile.nextLine();

        Scanner scLine = new Scanner(line).useDelimiter(",");

        nm = scLine.next();
        g = scLine.nextInt();
        i = scLine.next();
        h = scLine.nextDouble();
                                // onttrek velde

        if (i.equalsIgnoreCase(inC) || i.equals("Both"))
                                // kontroleer vir "Both" en kategorietipe
        {
            sSkik[sTelling] = new Student(nm, g, i, h);
                                // instansieer Student-skikkingsobjek
            sTelling++;          // inkrementeer sTelling
        }
    }
    scFile.close();
} catch (FileNotFoundException ex)
{
    System.out.println("File not found");
}
}

```

// Vraag 4.4 - 4

```

public String toString()
{
    String temp = "Naam:\t" + naam + "\nMaksimum:" + "\t" + maks + "\n";
    // korrekte opskrifinligting en formaat

    for (int i = 0; i < sTelling; i++) // lus deur skikking
    {
        temp += sSkik[i] + "\n";      // kombineer in 'n string
    }
    return temp;                      // stuur gekombineerde string terug
}
}

```

// Vraag 4.5 - 6

```

public void sort()
{
    for (int i = 0; i < sTelling - 1; i++) // korrekte buitelus
    {
        for (int j = i + 1; j < sTelling; j++) // korrekte binnelus
        {
            if (sSkik[i].kryGraad() > sSkik[j].kryGraad())
                                // korrekte if-stelling
                                // sorteer in korrekte
                                // volgorde
            {
                Student temp = sSkik[i];
                sSkik[i] = sSkik[j];
                sSkik[j] = temp;    // korrekte ruiling
            }
        }
    }
}

```

```

    }
  }
}

```

// Vraag 6 - 13

//4 punte vir 'n suksesvolle skraping - die skraping hoef nie in 'n afsonderlike metode te wees nie

```

private void skrapStudent(int inP)
{
    sSkik[inP] = sSkik[sTelling - 1];
    // korrekte kode om item te skrap
    sort();
    sTelling--;
    // dekrementeer sTelling
}

```

//ALTERNATIEF skrapStudent

```

private void skrapStudent(int inP)
{
    for (int i = inP; i < sTelling; i++)
    {
        sSkik[i] = sSkik[i+1]; // korrekte kode om item te skrap
    }
    sTelling--; // dekrementeer sTelling
}

```

// 9 punte om skikking te verwerk

```

public String korrekteGetalle()
{
    if (sTelling > maks) // kontroleer of items geskrap moet word
    {
        int getalGeskrap = sTelling - maks; // bepaal hoeveelheid om te
        // skrap
        String deletedList = "Studente verwyder:\n";
        for (int i = 0; i < getalGeskrap; i++) // lus deur skikking
        {
            int item = (int) (Math.random() * sTelling);
            // genereer lukraak getal binne bestek
            deletedList += sSkik[item] + "\n";
            // skep string met geskrapte items
            skrapStudent(item);
            // roep skrapmetode OF skrapkode word hier geplaas
        }
        return deletedList + "\nOorblywende studente:\n" + toString();
        // oorblywende string geskep beide stringe korrek met opskrifte
    }
}

```

// Vraag 7 - 9

//6 punte om GraadSkikking gSkik te instansieer

```

public String skepGraadSkikking(String inH) // korrekte opskrif
{

```

```

int graad;
double ure;

Scanner scLine = new Scanner(inH).useDelimiter(";");
while (scLine.hasNext())
{
    graad = scLine.nextInt();
    ure = scLine.nextDouble();    // velde korrek onttrek

    gSkik[graad - 8] = new Graad(graad, ure);
                                // gSkik korrek geïnstansieer
}
scLine.close();
return vertoonAlleGraadSkikking(); // stuur 'n string terug
}

```

//3 punte om 'n string van gSkik te skep - dit hoef nie 'n afsonderlike metode te wees nie

```

private String vertoonAlleGraadSkikking()
{
    String tempSt = "\nGraadtotale:\n"; // korrekte opskrif

    for (int i = 0; i < 5; i++)    // lus om Graad-skikking te verwerk
    {
        tempSt += gSkik[i] + "\n"; // objekte korrek gekombineer
    }
    return tempSt;
}

```

VRAAG 5 & 8 PROJEK GK-KLAS

// Vraag 5.1 - 1

```

public class ProjekGK // korrekte opskrif
{
    public static void main(String[] args)
    {
        // Vraag 5.2 - 1
        Projek klere = new Projek("Samel ou klere in","Binnenshuis",12);
        // projek korrek geïnstansieer

        // Vraag 5.3 - 2
        klere.sort();
        System.out.println(klere);
        // sorteerm metode geroep en objek vertoon

        // Vraag 8.1 - 1
        System.out.println(klere.korrekteGetalle());
        // roep korrekteGetalle korrek

        // Vraag 8.2 - 1
        System.out.println
(klere.skepGraadSkikking("9;53;12;13;8;72;11;90;10;34"));
        // roep skepGraadSkikking korrek
    }
}

```



```

    }
}

```

DELPHI-OPLOSSING

VRAAG 2 STUDENT-KLAS

```
unit uStudent;
```

```
interface
  uses SysUtils;
```

```
// Vraag 2.1 - 4
```

```
type TStudent = class      // klasopskrif
  private                 // alle kenmerke privaat
    volleNaam : string;   // almal korrek benoem
    graad : integer;      // korrekte tipes
    belangstelling : string;
    ure : double;
  public
    constructor Create(inFn: string; inG: integer; inI: string; inH: double);
    function kryGraad() : integer ;
    function kryBelangstelling(): string;
    function kryUre() : double;
    function veranderNaam() : string;
    function toString() : string;
end;
```

```
implementation
```

```
// Vraag 2.2 - 3
```

```
  constructor TStudent.Create(inFn: string; inG: integer; inI: string;
    inH: double);      // korrekte opskrif en parametername
  begin
    volleNaam := inFn; // velde tot parameters gestel
    graad := inG;
    belangstelling := inI;
    ure := inH;
  end;
```

```
// Vraag 2.3 - 2
```

```
  function TStudent.kryGraad() : integer;      // korrekte opskrifte vir al
                                                // drie metodes
  begin
    Result := graad;
  end;

  function TStudent.kryBelangstelling() : string; // korrekte terugsendings vir
                                                // al drie metodes
  begin
    Result := belangstelling;
  end;

  function TStudent.kryUre() : double;
```

```
begin
  Result := ure;
end;
```

// Vraag 2.4 - 4

```
function TStudent.veranderNaam() : string; // korrekte opskrif
var
  van, voorletter: string;
begin
  voorletter := volleNaam[1]; // isoleer voornaam & voorletter
  van := Copy(volleNaam, Pos(' ', volleNaam) +1, Length(volleNaam ));
  // isoleer van
  Result := van + ', ' + voorletter; // kombineer in die korrekte formaat
end;
```

// Vraag 2.5 - 3

```
function TStudent.toString() : string; // korrekte opskrif
var i : integer;
begin
  Result := veranderNaam() + #9 + IntToStr(graad) + #9 + belangstelling
    + #9 + floattostr(ure);
  // roep veranderNaam()
  // korrekte formaat met tabelkarakters
end;
end.
```

VRAAG 3 GRAAD-KLAS

```
unit uGraad;
```

```
interface
  uses SysUtils;
```

// Vraag 3.1 - 2

```
type TGraad = class // korrekte naam
  private
    graad : integer; // korrekte velde graad en totaal
    totaal : double;
```

// Vraag 3.2 - 2

```
  const
    PERK = 20; // korrekte waardes toegeken
    BONUS = 10; // konstante velde verklaar
  public
    constructor Create(inG: integer; inT: double);
    function kryTotaal() : double ;
    procedure stelTotaal(inT: double);
    function toString() : string;
end;
```

```
implementation
```

// Vraag 3.3 - 4

```
  constructor TGraad.Create(inG: integer; inT: double);
  begin
```

```

    graad := inG;                // ken graad-veld toe

    if (inT > PERK) then        // vergelyk inT met PERK
    begin
        totaal := inT + BONUS; // tel BONUS by totaal wanneer > PERK
    end
    else
    begin
        totaal := inT;         // andersins ken inT toe aan totaal
    end;
end;

```

// Vraag 3.4 - 2

```

function TGraad.kryTotaal() : double;
begin
    Result := totaal; // korrekte toegangermetode
end;

procedure TGraad.stelTotaal( inT: double); // korrekte mutatormetode
begin
    totaal := inT;
end;

```

// Vraag 3.5 - 2

```

function TGraad.toString() : string; // korrekte opskrif
var i : integer;
begin
    Result := 'Graad:' + IntToStr(graad) + ' totale ure '
        + floattostr(totaal); // velde korrek gekombineer
end;
end.

```

VRAAG 4, 6 & 7 PROJEK-KLAS

```
unit uProjek;
```

```

interface
uses
    SysUtils, uStudent, uGraad;

```

// Vraag 4.1 - 2

```

type TProjek = class // korrekte opskrif
private
    naam : string; // velde naam en maks korrek verklaar
    maks : integer;
    function vertoonAlleGraadSkikking() : string ;
public

```

// Vraag 4.2 - 4

```

    sSkik : array[1..50] of TStudent; // korrekte objeknaam
                                        // 50 elemente verklaar
    sTelling : integer; // sTelling korrek verklaar
    gSkik : array[1..5] of TGraad; // gSkik korrek verklaar
    constructor Create(inN, inC : string; inM: integer);
    function toString() : string;
    procedure sort();
    procedure skrapStudent(inP : integer);

```

```

    function korrekteGetalle() : string;
    function skepGraadSkikking(inH : string) : string;
end;
implementation

```

// Vraag 4.3 - 8

```

constructor TProjek.Create(inN, inC : string; inM: integer);
var
    inFile : TextFile;    // velde naam en maks korrek toegeken
    line : string;

    nm,i : string;
    g : integer;
    h : double;
begin
    naam := inN;
    maks := inM;
    sTelling := 0;

    AssignFile(inFile, 'Studente.txt'); // lêer oopgemaak om te lees
    Reset(inFile);

    while NOT EOF(inFile) do          // lus om lêer te lees
    begin
        Readln(inFile, line);

        nm := Copy(line, 1, Pos(',', line) -1 );
        Delete(line, 1, Pos(',', line));
        g := StrToInt(Copy(line, 1, Pos(',', line) -1));
        Delete(line, 1, Pos(',', line));
        i := Copy(line, 1, Pos(',', line) -1 );
        Delete(line, 1, Pos(',', line));
        h := strtofloat(line);

                                // onttrek velde

        if (CompareText(inC, i)=0) or (i='Both') then
                                // kontroleer vir "Both" en kategorietipe
        begin
            sTelling := sTelling + 1;    // inkrementeer sTelling
            sSkik[sTelling] := TStudent.Create(nm, g, i, h);
                                // instansieer Student-skikkingsobjek
        end;
    end;
end;
end;
end;

```

// Vraag 4.4 - 4

```

function TProjek.toString() : string;
var i : integer;
begin
    Result := 'Naam: ' + #9 + naam + #10#13 + 'Maksimum:' + #9 + IntToStr(maks)
        + #13#10; // korrekte opskrifinligting en formaat
    for i:= 1 to sTelling do // voer lus uit deur skikking
        Result := Result + sSkik[i].toString() + #13#10;
                                // kombineer in 'n string
                                // stuur gekombineerde string terug
    end;
end;

```

// Vraag 4.5 - 6

```

procedure TProjek.sort() ;
var
  i, j : integer;
  temp : TStudent;
begin
  for i:= 1 to sTelling do      // korrekte buitelus
    for j:= 1 to sTelling do   // korrekte binnelus
      begin
        if (sSkik[i].kryGraad() < sSkik[j].kryGraad()) then
          begin
            // korrekte if-stelling
            // sorteer in korrekte volgorde
            temp := sSkik[i];
            sSkik[i] := sSkik[j];
            sSkik[j] := temp;      // korrekte ruiling
          end;
        end;
      end;
    end;
  end;
end;

```

// Vraag 6 - 13

//4 punte vir 'n suksesvolle skrapping - die skrapping hoef nie in 'n afsonderlike metode te wees nie

```

procedure TProjek.skrapStudent(inP: Integer);
begin
  sSkik[inP] := sSkik[sTelling - 1];
  sort();          // korrekte kode om item te skrap

  sTelling := sTelling -1; // dekrementeer sTelling
end;

```

//ALTERNATIEF skrapStudent

```

procedure TProjek.skrapStudent(inP: Integer);
begin
  for i:= inP to sTelling do
    sSkik[i] := sSkik[i+1]; // korrekte kode om item te skrap
  sTelling := sTelling -1; // dekrementeer sTelling

end;

```

// 9 punte om skikking te verwerk

```

function TProjek.korrekteGetalle() : string;
var
  getalGeskrap, i, randPos : integer;
  deletedList : string;
begin
  if (sTelling > maks) then      // kontroleer of items geskrap moet word
  begin
    getalGeskrap := sTelling - maks; // bepaal hoeveelheid om te skrap
    deletedList := 'Studente verwyder:' + #13#10;

    for i := 1 to getalGeskrap do // lus deur skikking

      begin
        randPos := Random(sTelling)+1;
        // genereer lukraak getal binne bestek

```

```

    deletedList := deletedList + sSkik[randPos].toString() + #13#10;
                    // skep string met geskrapte items

    skrapStudent(randPos);
                    // roep skrapmetode OF skrapkode word hier geplaas
end;
Result := deletedList + #13#10 + 'Oorblywende studente:' + #13#10
        + toString();
// oorblywende string geskep beide stringe korrek met opskrifte

end;

end;

```

// Vraag 7 - 9
//6 punte om GraadSkikking gSkik te instansieer

```

function TProjek.skepGraadSkikking(inH : string) : string;
                    // korrekte opskrif

var
    graad : integer;
    ure : double;
begin
    while Pos(';', inH) > 0 do
        begin
            graad := StrToInt(Copy(inH, 1, Pos(';', inH) -1 ));
            Delete(inH, 1, Pos(';', inH));
            if (Pos(';', inH) <> 0) then
                begin
                    ure := StrToFloat(Copy(inH, 1, Pos(';', inH) -1));
                    Delete(inH, 1, Pos(';', inH));
                end
            else ure:= StrToFloat(inH);
                    // velde korrek onttrek

            gSkik[graad-7] := TGraad.Skep(graad, ure);
                    // gSkik korrek geïnstansieer
        end;
        Result := vertoonAlleGraadSkikking; // stuur 'n string terug
    end;

```

//3 punte om 'n string van gSkik te skep - dit hoef nie 'n afsonderlike metode te wees nie

```

function TProjek.verteonAlleGraadSkikking() : string;
var
    i : integer;
begin
    Result := #13#10 + 'Graadtotale:' + #13#10; // korrekte opskrif

    for i:=1 to 5 do
        Result := Result + gSkik[i].toString() + #13#10;
        // lus om Graad-skikking te verwerk
        // objekte korrek gekombineer
    end;

end.

```

VRAAG 5 & 8 PROJEK GK-KLAS**// Vraag 5.1 - 1**program ProjekGK; **// korrekte opskrif**

{\$APPTYPE CONSOLE}

{\$R *.res}

uses

System.SysUtils,
uStudent in 'uStudent.pas',
uProjek in 'uProjek.pas',
uGraad in 'uGraad.pas';

var

klere : TProjek;
temp : string;

begin

try

// Vraag 5.2 - 1

klere := TProjek.Skep('Samel ou klere in', 'Binnenshuis', 12);

// projek korrek geïnstansieer**// Vraag 5.3 - 2**

klere.sort();

WriteLn(klere.toString());

// sorteerm metode geroep en objek vertoon**// Vraag 8.1 - 1**

WriteLn(klere.korrekteGetalle());

// roep korrekteGetalle korrek**// Vraag 8.2 - 2**

WriteLn(klere.skepGraadSkikking ('9;53;12;13;8;72;11;90;10;34'));)

// roep skepGraadSkikking korrek

ReadLn(temp);

except

on E: Exception do

WriteLn(E.ClassName, ': ', E.Message);

end;

end.

AFVOER**AFDELING A****VRAAG 1.1**

StudentID	StudentNaam	IsSenior
53	Amy Radebe	TRUE
30	Andrea Badenhorst	TRUE
54	Brendan Smit	TRUE
1	Jacob Ncube	TRUE
25	Joshua Jacobs	TRUE
22	Julia Hudson	TRUE
3	Karabo Mlangeni	TRUE
47	Kendal Buys	TRUE
12	Kenneth Motala	TRUE
17	Kobus Venter	TRUE
37	Laetitia Adams	TRUE
38	Lesego Semenya	TRUE
11	Michael Stemmet	TRUE
49	Mikyle Sithole	TRUE
43	Mthokozisi Kumalo	TRUE
39	Nina Ntsimango	TRUE
9	Patricia Davids	TRUE
16	Penny Mbele	TRUE
44	Prince Dube	TRUE
20	Rethabile Mokone	TRUE
46	Sego Dlamini	TRUE
27	Somizi Baloyi	TRUE
8	Steven Govender	TRUE
51	Tasneem Morkel	TRUE
33	Vicki de Beer	TRUE
41	Wian Oosthuizen	TRUE
18	Wiseman Legodi	TRUE

VRAAG 1.2

ProjekNaam
SAT English tutoring program
FRI Mathematics tutorials

VRAAG 1.3 *Data kan anders geformateer wees op jou rekenaar*

RegistrasielD	Expr1001
1	1.0833333333333333
2	0.8928571428571429
3	0.8
4	0.7857142857142857
5	0.84375
6	1.09375
7	0.7083333333333334
8	0.9038461538461539
9	0.925
10	0.9583333333333334
11	0.9318181818181818
12	1.0666666666666667
13	1.125
14	0.8125
15	1.0
16	1.0

Vervolg op die volgende bladsy

RegistrasielD	GemUre
17	0.875
18	0.5
19	0.8333333333333334
20	1.125
21	0.8846153846153846
22	1.5
23	0.9318181818181818
24	1.5
25	0.9090909090909091
26	0.5
27	0.8636363636363636
28	1.0
29	0.5
30	1.5
31	0.5
32	0.75
33	0.8
34	0.95
35	0.5
36	1.5
37	0.8
38	1.5
39	1.0555555555555556
40	0.7
41	1.0
42	1.5
43	0.8
44	0.9444444444444444
45	0.75
46	1.0
47	0.5
48	1.0
49	0.6
50	1.0
51	1.5
52	0.5
53	0.8333333333333334
54	0.5
55	1.5
56	1.2
57	0.9444444444444444
58	0.9
59	0.875
60	0.875
61	0.9166666666666666
62	0.8125
63	0.9375
64	0.96875
65	1.5
66	0.5
67	0.9285714285714286
68	0.9285714285714286
69	1.0
70	0.8125
71	0.5
72	0.5
73	1.5

VRAAG 1.4

Dag
FRI
MON
SAT
THU
TUE
WED

VRAAG 1.5

ProjekNaam	StudentNaam
THU Sandwich feeding scheme	Jacob Ncube
SAT English tutoring programme	Karabo Mlangeni
SAT River clean up	Penny Mbele
FRI Mathematics tutorials	Kenneth Motala
TUE Knitting for moms	Patricia Davids
MON Old Age Home Visit	Steven Govender
WED Recycling Programme	Wiseman Legodi
FRI Textbook Collection	Joshua Jacobs

VRAAG 1.6

StudentID	TotaalUre
1	13,5
2	16,25
3	8,5
4	17,5
5	9,25
6	13,5
7	10,5
8	7
9	5,5
10	11,5
11	19,5
12	10,25
13	14
14	16,5
15	14,5
16	10
17	18,25
18	6,5
19	7
20	3,75
21	10
22	12,5
23	11,75
24	10
25	15
26	11,5
27	6
28	4
29	17,5
30	11
31	14,5
32	7,5
33	15
35	15
36	6
37	16
38	8

StudentID	TotaalUre
39	9,5
40	8,5
41	10
42	15
43	10,75
44	6
45	8
46	13
47	6
49	10,5
50	9
51	12,5
52	4,5
53	8
54	8,25
55	13

VRAAG 1.7

StudentNaam
Khaya Mokoena
Mishka Hassen

VRAAG 1.8

ProjekNaam
TUE Knitting for moms
FRI Textbook Collection

VRAAG 1.9

StudentNaam	GetalProjekte
Bhule Mbasa	2
Blessing Mkhize	2
Conrad Snyman	2
Frans Theron	2
Heinriche Pretorius	2
Joshua Jacobs	2
Kobus Venter	2
Laetitia Adams	2
Mahmood Chetty	3
Mary-anne Muir	2
Michael Stemmet	2
Mikyle Sithole	2
Mthokozisi Kumalo	2
Nikita van Wyk	2
Nina Ntsimango	2
Patience Madonsela	2
Sego Dlamini	2
Tasneem Morkel	2
Vicki de Beer	2

AFDELING B**FINALE AFVOER**

Naam: Samel ou klere in

Maksimum:	12		
Pettie, F	8	Binnenshuis	2.5
Maler, S	8	Albei	0.0
Honiford, E	8	Binnenshuis	5.5
Doyle, J	9	Albei	4.0
Boyder, L	9	Albei	9.5
Delaney, B	9	Binnenshuis	3.0
Leaby, T	9	Albei	7.0
Shorts, H	9	Albei	12.0
Rabey, C	10	Binnenshuis	4.0
Janson, C	10	Albei	7.0
Scotty, M	10	Binnenshuis	1.0
Leaby, G	10	Binnenshuis	3.0
Morvel, M	11	Albei	1.0
Heriot, L	11	Binnenshuis	3.5
McCalum, A	11	Albei	5.0
Monahan, B	12	Binnenshuis	5.0

Studente verwyder:

// hierdie lys sal varieer na gelang van watter student ewekansig geselekteer is

Morvel, M	11	Albei	1.0
Leaby, T	9	Albei	7.0
Boyder, L	9	Albei	9.5
Doyle, J	9	Albei	4.0

Oorblywende studente:

// hierdie lys sal varieer na gelang van watter student ewekansig geskrap is

Naam: Samel ou klere in

Maksimum:	12		
Pettie, F	8	Binnenshuis	2.5
Maler, S	8	Albei	0.0
Honiford, E	8	Binnenshuis	5.5
Delaney, B	9	Binnenshuis	3.0
Shorts, H	9	Albei	12.0
Rabey, C	10	Binnenshuis	4.0
Janson, C	10	Albei	7.0
Scotty, M	10	Binnenshuis	1.0
Leaby, G	10	Binnenshuis	3.0
Heriot, L	11	Binnenshuis	3.5
McCalum, A	11	Albei	5.0
Monahan, B	12	Binnenshuis	5.0

Graadtotale:

Graad:8	totale ure	82.0
Graad:9	totale ure	63.0
Graad:10	totale ure	44.0
Graad:11	totale ure	100.0
Graad:12	totale ure	13.0