

UITWERKINGEN

Opgave 1.

a) 1 - 3 - 1
 2 - 3 - 2
 2 - 3 - 2
 2 - 6 - 2
 4 - 6 - 4
 4 - 3 - 4
 4 - 3 - 4
 4 - 2 - 4

b) 1- range(1, 6)
 2- range(10, 22, 3)
 3- range(12, -4, -4)

 4- '4444'
 5- '5678'
 6- '4444'

c) def bereken (x, n):
 resultaat =0.0
 macht = x * x
 teken = 1
 factor = 5
 faculteit = 1

 for i in range(n):
 term = teken * factor * macht / faculteit
 resultaat += term

 teken = -teken
 factor += 1
 macht *= x
 faculteit *= (i+2)

 return resultaat

d) -5 6
 12 7
 -5 -4
 -4 -4
 4 0
 -3 -4
 -2 -1

Opgave 3.

```
def verschuif_matrix (m):  
    for rij in m:  
        verschuif(rij)  
  
def verschuif (rij):  
    int_in_eerste_kolom = rij[0]  
    k = len(rij)  
    for i in range(1, k):  
        rij[i-1] = rij[i]  
    rij[k-1] = int_in_eerste_kolom
```

#alternatieve oplossing

```
def verschuif_matrix (m):  
    for i in range(len(m)):  
        m[i] = m[i][1:] + [m[i][0]]
```

Opgave 2.

- a) Voeg toe aan de class Universiteit.

```
def voeg_toe (self, a_student):  
    self.student_list.append(a_student)
```

- b) Voeg toe aan de class Student.

```
TOPSTUDENT_GRENS = 9.0
```

```
def gemiddelde (self):  
    totaal = 0.0  
    for vak in self.vakken:  
        totaal += vak.cijfer  
    return totaal / len(self.vakken)
```

```
def is_topstudent (self):  
    return self.gemiddelde() >= Student.TOPSTUDENT_GRENS
```

Voeg toe aan de class Universiteit.

```
def topstudenten (self):  
    resultaat = Universiteit()  
  
    for student in self.student_list:  
        if student.is_topstudent():  
            resultaat.voeg_toe(student)  
  
    return resultaat
```

- c) Voeg toe aan de class Vak.

```
VOLDOENDE_GRENS = 5.5
```

```
def is_voldoende (self):  
    return self.cijfer >= Vak.VOLDOENDE_GRENS
```

Voeg toe aan de class Student.

```
GOED_PERCENTAGE = 95
```

```
def percentage_voldoendes (self):  
    aantal_voldoendes = 0  
    for vak in self.vakken:  
        if vak.is_voldoende():  
            aantal_voldoendes += 1  
    return 100 * float(aantal_voldoendes) / len(self.vakken)
```

```
def doet_het_goed (self):  
    return self.percentage_voldoendes() >= Student.GOED_PERCENTAGE
```

Voeg toe aan de class Universiteit.

```
def gaat_goed (self):  
    int resultaat = 0  
  
    for student in self.student_list:  
        if student.doet_het_goed():  
            resultaat += 1  
  
    return resultaat
```

d) Voeg toe aan de class Universiteit.

```
def gaat_goed_en_ingeschreven (self, jaar):  
    return self.ingeschreven(jaar).gaat_goed()
```

e) Voeg toe aan de class Student.

```
def verwijder_vak (self, naam):  
    for vak in self.vakken:  
        if vak.naam == naam:  
            self.vakken.remove(vak)  
    return
```

Voeg toe aan de class Universiteit.

```
def verwijder (self, a_str1, a_str2):  
    for student in self.student_list:  
        if student.naam == a_str1:  
            student.verwijder_vak(a_str2)  
    return
```