|  |
| --- |
| **Toets Informatica**  **Programmeren met Python III**    Stof:   * Basiskennis van Python * (Eigen) Functies & parameters * Tekstbestanden * Strings * Geneste loops en geneste lijsten   Schrijf je antwoorden op een **apart papier**. Een gewone (niet-grafische) rekenmachine is toegestaan.  Deze toets bestaat uit **XX opgaven**. **Norm**: Bij elke vraag staat aangegeven hoeveel punten je maximaal voor die vraag kunt behalen. **Totaal YY punten.**  **Succes!** |

## [2pt] Opgave 1

Geef voor ieder van de volgende beweringen aan of deze WAAR of ONWAAR is:

1. Als je een bestand opent om te lezen en het bestand bestaat nog niet, dan zal een nieuw bestand aangemaakt worden.
2. Als je een bestand opent om in te schrijven en het bestand bestaat nog niet, dan zal een nieuw bestand aangemaakt worden.
3. Als je een bestand opent om in te schrijven en het bestand bestaat al, dan zal over het oorspronkelijke bestand overschreven worden.

## [2pt] ANTWOORD Opgave 1

1. FALSE: Het programma geeft een fout: FileNotFoundError [Errno 2] No such file or directory: naam bestand
2. TRUE
3. TRUE

[-1pt per fout, max 2pt aftrek]

## [1pt] Opgave 2

Om een bestand als geheel in te lezen als één string gebruiken we:

1. bestand.read()
2. bestand.readline()
3. bestand.readlines()
4. open(bestandsnaam, “r”)

## [1pt] ANTWOORD Opgave 2

a

## [2pt] Opgave 3

De volgende vraag gaat over de code hieronder. Je mag ervan uitgaan dat getallen een lijst met gehele getallen is.

|  |
| --- |
| def functie1(getallen):  antwoord = [ ]  for getal in getallen:  if getal > 0:  antwoord = antwoord + [getal]  return antwoord |

Vat kort samen wat het doel van ‘functie1’ is. Let op: Er wordt gevraagd om een samenvatting, dus niet een uitleg van elke regel.

## [2pt] ANTWOORD Opgave 3:

De functie levert een lijst [1pt] op met de positieve getallen uit de oorspronkelijke lijst [2pt].

of

De functie levert een lijst [1pt] op met alleen de getallen groter dan 0 uit de oorspronkelijke lijst [2pt].

Omdat het qua syntax veel op som van getallen lijkt (als je [ ] over het hoofd ziet), wordt maar minimale aftrek hiervoor gegeven: -1pt

## [4pt] Opgave 4

De volgende vraag gaat over de code hieronder. Je mag ervan uitgaan dat getallen een lijstje getallen is.

|  |
| --- |
| def functie2():  a = getallen[0]  for waarde in getallen:  if waarde > a:  a = waarde  return a  def functie3(a):  b = 0  for waarde in getallen:  if waarde == a:  b += 1  return b  c = functie2()  d = functie3(c)  print(d) |

a) Wat is de uitvoer van ‘functie2’ als getallen = [2,-4,7,4,7]

b) Vat kort samen wat het doel van het hele programma is. Let op: Er wordt gevraagd om een samenvatting, dus niet een uitleg van elke regel.

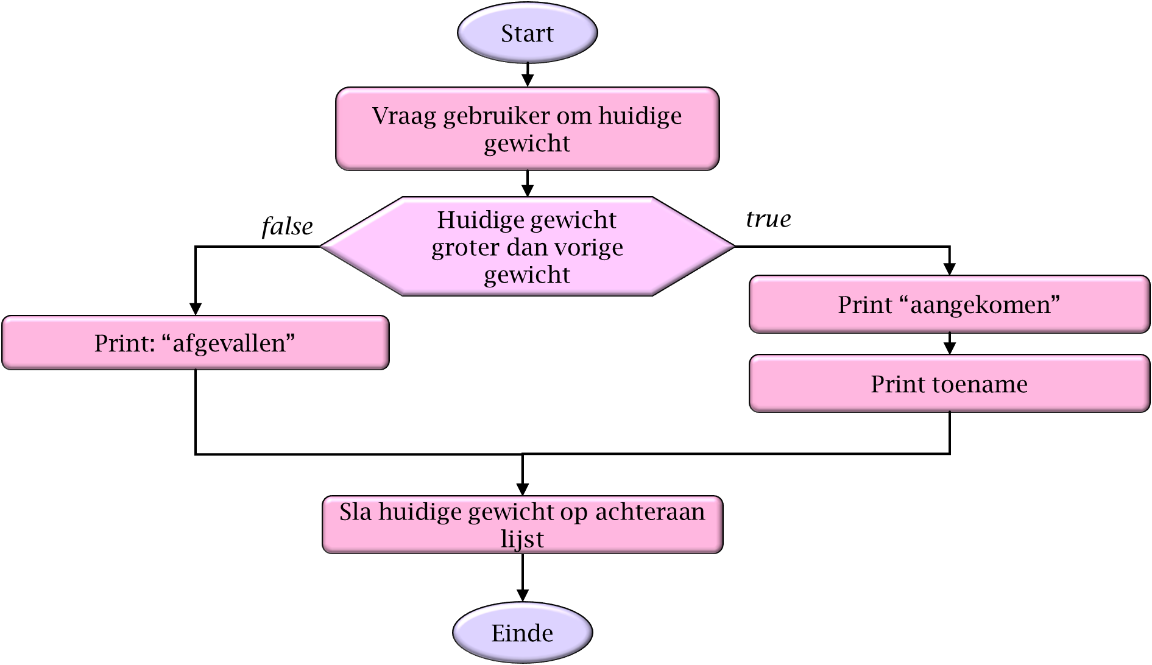
Tips: Ga na wat het doel van ‘functie2’ is. Ga na wat het doel van ‘functie3’ is. Wat is hun doel samen als geheel?

[4pt] ANTWOORD Opgave 4:

1. [1pt] 7 (want functie2 bepaalt het maximum)
2. [3pt] Het programma drukt af hoe vaak [1pt] het grootste getal [1pt] voorkomt in een niet-lege lijst[1pt].

## [4pt] Opgave 5

Een atleet schrijft een programma om zijn gewicht (in kilo’s) bij te houden. Voor het onthouden wordt een lijst gebruikt: telkens wordt het gewicht achteraan de lijst toegevoegd. Hieronder zie je een stroomdiagram voor dat programma. Vertaal het stroomdiagram naar Pythoncode. Je kunt uitgaan van deze lijst met de gewichten van drie metingen: gewichtenLijst = [56.3, 54.3, 58.9]. Het laatst gemeten gewicht kun je dus uit de lijst opvragen.



## [4pt] ANTWOORD Opgave 5:

gewichtenLijst = [56.3, 54.3, 58.9]  
  
nieuwGewicht = float (input (**"Hoeveel weeg je?"**) )  
  
**if** nieuwGewicht > gewichtenLijst[ len(gewichtenLijst)-1 ]:

*# OF*

*#vorigeGewicht = gewichtenLijst[len(gewichtenLijst)-1]*

*#if (nieuwGewicht - vorigeGewicht) > 0:*

print(**"Je bent aangekomen"**)  
 aangekomen = nieuwGewicht - gewichtenLijst[ len(gewichtenLijst)-1 ]  
 print(aangekomen)  
  
**else**:  
 print(**"Je bent afgevallen"**)

gewichtenLijst += [nieuwGewicht]  
*# OF  
#gewichtenLijst.append(nieuwGewicht)  
  
#om te testen:  
#print(gewichtenLijst)*

[-1pt voor elke fout: ontbreken van opslaan gebruikersinvoer in variabele, float, if, len, len-1, berekening, juiste prints, opslaan in lijst]

## [3pt] Opgave 6

De volgende vraag gaat over de code hieronder.

|  |
| --- |
| def functie4(val):    for i in range(val):      for j in range(i):        print("\*", end="")      print() |

Vat kort samen wat het doel van ‘functie4’ is. Let op: Er wordt gevraagd om een samenvatting, dus niet een uitleg van elke regel.

## [3pt] ANTWOORD Opgave 6:

* Print een driehoekig vak met sterretjes: 4 rijen, met eerste rij 0 sterren, dan 1 … 3
* 1pt aftrek bij vierkant of ander patroon
* 1pt aftrek per verkeerd/ontbrekend stopcriterium

## [1pt] Opgave 7

Wat tekent de onderstaande code? De turtle begint altijd gericht naar rechts.

|  |  |
| --- | --- |
| **import** turtle  turtle.pendown() **for** x **in** range(4):  **for** y **in** range (360):  turtle.forward(1)  turtle.right(1)  turtle.penup()  turtle.forward(100)  turtle.pendown() | 1. 4 cirkels naast elkaar 2. 4 cirkels onder elkaar 3. 360 vierkanten 4. 4 vierkanten onder elkaar |

## [1pt] Antwoord Opgave 7:

a. 4 cirkels naast elkaar

## [4pt] Opgave 8:

Schrijf een programma dat, gegeven een lijst met getallen, een staafdiagram met sterretjes afdrukt. Bijvoorbeeld bij [1,3,0,2,4] wordt afgedrukt:

\*

\*\*\*

\*\*

\*\*\*\*

## [4pt] ANTWOORD OPGAVE 8

lijst = [1,3,0,2,4]

for getal in lijst:

  for i in range(getal):

    print("\*", end="")

  print()

OF:

for getal in lijst:

  aantal\_sterren = 0

  while aantal\_sterren < getal:

    print("\*", end="")

    aantal\_sterren+=1

  print()

OF:

for x in lijst:

print(x\*"\*", end="")

print()

OF:

for y in (lijst):

  aantal\_ster = y \* '\*'

  print(aantal\_ster)

OF:

for i in lijst:

print(i \* "\*")

[1pt aftrek voor elke fout]

## [7pt] Opgave 9

|  |
| --- |
| woord = (“fiets\nbel”)  print(woord) |

1. [1pt] Wat wordt er afgedrukt?
2. [1pt] Wat levert woord.find(“ts”) op?
3. [1pt] Hoe druk je de eerste letter van woord af?
4. [2pt] Zonder gebruik te maken van een cijfer, hoe druk je de laatste letter van woord af? Je mag in je antwoord de getallen van 0 t/m 9 dus niet gebruiken.
5. [2pt] Schrijf een functie die woord als parameter binnenkrijgt en het woord in hoofdletters oplevert.

## [7pt] ANTWOORD Opgave 9

1. [1pt] Over twee regels afgedrukt:

fiets

bel

1. [1pt] 3
2. [1pt] print( woord[0] )
3. [2pt] print( woord[len(woord)-1] )

[1pt aftrek per fout]

1. [2pt]

def maakHoofdletters(woord):

hoofdletter\_woord = woord.upper()

return hoofdletter\_woord

OF:

def kapitalen(woord):

return woord.upper()

[1pt aftrek per fout]

## [6pt] Opgave 10

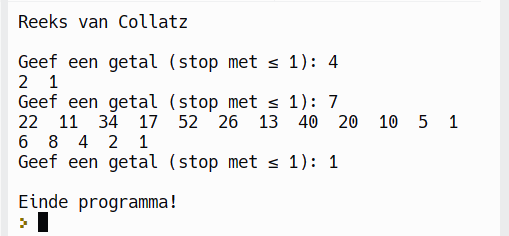
Hieronder zie je het stroomdiagram voor de ***reeks van Collatz***:



Schrijf een programma waarin:

1. [1pt] een functie staat die bepaalt of een getal even is;
2. [1pt] eerst om een getal als invoer wordt gevraagd;
3. [3pt] de reeks van Collatz wordt opgeleverd met de getallen **achter elkaar** door spaties gescheiden;
4. [1pt] na het presenteren van de getallenreeks opnieuw om een getal gevraagd wordt. Stop als de invoer 1 of kleiner dan 1 is.

Voorbeeld:



## [6pt] ANTWOORD Opgave 10

def getalIsEven(getal): # onderdeel a)

if getal % 2 == 0:

return True

else:

return False # einde functie getalIsEven onderdeel a)

# hoofdprogramma

nieuweInput = True # Boolean flag voor inputvraag onderdeel b)

print("Reeks van Collatz") # titel

print()

while nieuweInput:

getal = int(input("Geef een getal (stop met \u2264 1): ")) # einde onderdeel b)

# (\u2264 = ≤)

if getal <= 1: # onderdeel d)

nieuweInput = False

else: # einde onderdeel d)

while getal != 1: # onderdeel c)

while getal != 1:

if getalIsEven(getal): # call onderdeel a)

getal /= 2

else:

getal \*= 3

getal += 1

print(int(getal), " ", end="")

print() # einde onderdeel c)

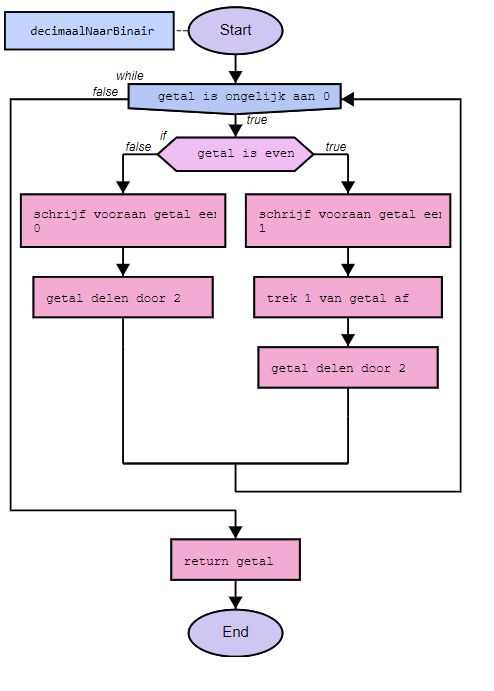
print("Einde programma!")

Voor ieder werkend onderdeel a), b) en d) van de opgave: +1 pt.

Onderdeel c): max 3 pt: per fout of vergeten statement: -1 pt met ondergrens 0 pt.

## [5pt] Opgave 11

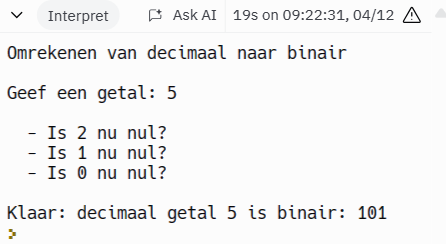
Hieronder zie je het stroomdiagram voor het omrekenen van een decimaal naar een binair getal:



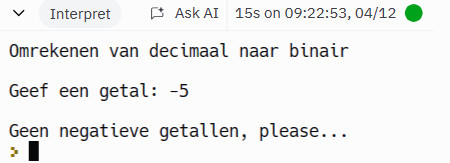
Schrijf het Python-programma hierbij.

1. [1pt] Vraag de gebruiker om een getal;
2. [1pt] Check of het groter dan of gelijk aan 0 is;
3. [1pt] Maak een functie die bepaalt of een getal even is. Gebruik hiervoor het bovenstaande stroomdiagram.
4. [1pt] Print telkens*: is het getal x nu nul*?
5. [1pt] Geef het eindresultaat: “*Klaar: decimaal getal x is binair y”*, waarbij x de invoer is en y het binaire getal.

Bij invoer van 5 moet het volgende verschijnen:



Bij -5:



## [6pt] ANTWOORD Opgave 11

def getalIsEven(getal): # onderdeel c)

if getal % 2 == 0:

return True

else:

return False # einde functie getalIsEven # einde onderdeel c)

# hoofdprogramma

print("Omrekenen van decimaal naar binair")

print()

binair = ""

getal = int(input("Geef een getal: ")) # onderdeel a)

print()

bewaarGetal = getal # einde onderdeel a)

if getal < 0: # onderdeel b)

print("Geen negatieve getallen, please...")

else:

while getal != 0:

if getalIsEven(getal): # call onderdeel c)

binair = "0" + binair

else:

binair = "1" + binair

getal -= 1

getal /= 2

print(" - Is", int(getal), "nu nul? ") # onderdeel d)

print()

print("Klaar: decimaal getal", bewaarGetal, "is binair: " + binair) # onderdeel e)

Voor ieder werkend onderdeel van de opgave: +1 pt.

## [8pt] Opgave 12

Bekijk onderstaande figuur goed. Ontwerp een algoritme waarin je eerst de grootste veelhoek tekent, daarna de op een na grootste en zo verder totdat de driehoek getekend is. Uiteindelijk moet de gebruiker zelf het aantal veelhoeken kunnen opgeven.

Afbeelding met schets, diagram, origami

Automatisch gegenereerde beschrijving

Neem de volgende overwegingen mee in je programma:

* Als je een algemeen te gebruiken veelhoek wil tekenen, bedenk dan dat de grootte van de hoek die je moet draaien afhankelijk is van het aantal zijden dat de veelhoek heeft: in een vierhoek draai je hoeken van 90 graden; in een vijfhoek van 72 graden; in een zeshoek van 60 graden.
* Als je vier veelhoeken wil tekenen, moet de grootste veelhoek zes zijden hebben. Er is dus een afhankelijkheid tussen aantal veelhoeken en aantal zijden.

Maak het programma in de onderstaande stapjes:

1. [2pt] Teken een zeshoek met zijde gelijk aan 100 met behulp van een loop waarin de eerste overweging opgenomen is.
2. [3pt] Zet hier een loop omheen om vier veelhoeken te kunnen tekenen. Teken deze van groot naar klein. Zet het aantal van vier op een variabele aantalFiguren.
3. [1pt] Laat nu de gebruiker zelf het aantal veelhoeken invullen.
4. [1pt] Verzin iets voor het geval dat de gebruiker een getal kleiner dan 3 invult.
5. [1pt] Wat was de invoer voor deze figuur?

Afbeelding met schets, tekening, ontwerp

Automatisch gegenereerde beschrijving

## [8pt] Antwoord Opgave 12

1. [2pt]

import turtle

zijde = 100

for j in range(6):

hoek = 360/6 # dit is de eerste overweging [1pt]

turtle.forward(zijde)

turtle.left(hoek)

1. [3pt] **Beoordeel alleen de geel gemarkeerde regels**

import turtle

zijde = 100

aantalFiguren = 4

# drie regels om netjes te positioneren: niet verplicht

turtle.penup()

turtle.setposition(-100,-150)

turtle.pendown()

for i in range(aantalFiguren):

aantalZijden = aantalFiguren + 2 # dit is de tweede overweging [1pt]

for j in range(aantalZijden):

hoek = 360/(4+2) # of 6 i.p.v. 4+2

turtle.forward(zijde)

turtle.left(hoek)

aantalFiguren -=1 # regel in for i-loop: [1pt]

1. [2pt] **Beoordeel alleen de geel gemarkeerde regels**

import turtle

zijde = 100

aantalFiguren = int(input("Hoeveel veelhoeken wil je tekenen? "))

# drie regels om netjes te positioneren: niet verplicht

turtle.penup()

turtle.setposition(-100,-150)

turtle.pendown()

for i in range(aantalFiguren):

aantalZijden = aantalFiguren + 2

for j in range(aantalZijden):

hoek = 360 / aantalZijden

turtle.forward(zijde)

turtle.left(hoek)

aantalFiguren -= 1

1. [1pt] **Beoordeel alleen de geel gemarkeerde regels**

import turtle

zijde = 100

aantalFiguren = int(input("Hoeveel veelhoeken wil je tekenen? "))

if aantalFiguren > 2:

# drie regels om netjes te positioneren: niet verplicht

turtle.penup()

turtle.setposition(-100,-150)

turtle.pendown()

for i in range(aantalFiguren):

aantalZijden = aantalFiguren + 2

for j in range(aantalZijden):

hoek = 360 / aantalZijden

turtle.forward(zijde)

turtle.left(hoek)

aantalFiguren -= 1

else:

print("Een", str(aantalFiguren) + "-hoek is geen veelhoek.") # o.i.d.

1. [1pt] Wat was de invoer voor deze figuur? 12 (tel de veelhoeken)

Totaal: XX punten en YY opgaven (zoek in Word op: “] ANTWOORD”

Vergeet niet om XX te vervangen door het werkelijke aantal opgaven en scorepunten.