Let op: Bij deze toets ligt de nadruk op de laatste hoofdstukken 5 t/m 10. Zie Python toets I voor vragen gericht op hoofdstukken 1 t/m 7.

|  |
| --- |
| **Toets Informatica**  **Programmeren met Python II**  Stof: Hoofdstukken 1 t/m 10, (Eigen) Functies, Loops (for en while), Turtle graphics (loops en functies & parameters), Lijsten  Schrijf je antwoorden op een **apart papier.**  Je mag geen gebruik maken van een computer of een rekenmachine.  Deze toets bestaat uit XX **opgaven.** Bij elke vraag staat aangegeven hoeveel punten je maximaal voor die vraag kunt behalen. **Totaal XX punten.**  **Succes!** |

## [2pt] Opgave 1:

|  |
| --- |
| teller = 1  while teller **€** 10:    print("hallo")    teller += 2 |

In de conditie van de while staat een vierkantje. Geef voor elk van de volgende mogelijkheden aan hoe vaak “hallo” afgedrukt zal worden als we het vierkantje vervangen door een van de volgende tekens:

1. <=
2. >
3. =
4. !=

## [3pt] Opgave 2:

Schrijf het volgende programma om zodat het een **while**-loop gebruikt in plaats van een **for**-loop.

|  |
| --- |
| for x in range(1,3):    print(x)  print("klaar") |

## [2pt] Opgave 3:

Bekijk de volgende code:

|  |
| --- |
| **def** piep ( a ):  b = **"Hallo"** print( b )  **if** a > 4:  **return True  else**:  **return False**  piep(-12.2) |

Geef aan wat het **type** is van:

1. de parameter van de functie piep
2. wat er opgeleverd wordt
3. wat er afgedrukt wordt

[3pt] Opgave 4: We willen een programma schrijven dat de gebruiker om gehele getallen vraagt en deze bij elkaar optelt. Het programma stopt als de gebruiker een 0 ingeeft.

Geef aan welke van de volgende instructies nodig zijn, en in welke volgorde. Je hoeft niet alle regels te gebruiken, en als het nodig is mag je regels vaker gebruiken.

1. totaal = totaal + invoer
2. invoer = int( input (**"Voer een getal in: "**) )
3. invoer = str( input (**"Voer een getal in: "**) )
4. invoer = input (**"Voer een getal in: "**)
5. **while** invoer==0:
6. **while not** invoer==0:
7. totaal == 0
8. totaal = 0
9. print( totaal )
10. print = int( input (**"Voer een getal in: "**) )

[5pt] Opgave 5: Van de afgelopen dagen is de temperatuur van de koeling in een cafetaria bijgehouden. Hier zie je een voorbeeld van een lijst met metingen:

lijst = [3,8,7,6,4]

Schrijf een programma dat telt hoe vaak de temperatuur 7 graden of hoger is geweest, en dit afdrukt. Dit is een voorbeeld lijst, jouw programma moet natuurlijk voor elke lijst werken, met de naam ‘lijst’. Als er geen waarden in de lijst staan, dan moet je programma de tekst “Fout” printen.

## [2pt] Opgave 6:

Je wilt de getallen 1 tot en met 4 bij elkaar optellen. Een van de volgende codes is juist. Welke? Tip: maak een trace table waarin je bijhoudt wat de waarde van de variabelen gebeurt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | som = 0 **for** i **in** range(1,5):  som = som + 1 |  | **C** | **for** i **in** range(1,5):  som = 0  som = som + som |
| **B** | som = 0 **for** i **in** range(1,5):  som = som + i |  | **D** | **for** i **in** range(1,5):  som = 0  som = som + i |

## [3pt] Opgave 7:

|  |
| --- |
| getal = 6  def doeIets(n):  a = 0  b = 1    while b <= n:      a += b      b = b + 1    print (a)  doeIets(getal) |

Vat in een paar woorden duidelijk samen wat het doel van het programma is. Let op: er wordt *niet* gevraagd voor een regel-voor-regel uitleg.

Tip: maak eventueel een trace table.

## [3pt] Opgave 8:

De code hieronder tekent een rode driehoek:

|  |
| --- |
| **import** turtle  turtle.pendown() turtle.pencolor(**"red"**) turtle.forward(50) turtle.right(120) turtle.forward(50) turtle.right(120) turtle.forward(50)  turtle.done() |

Schrijf een functie voor het tekenen van een driehoek die twee parameters meekrijgt: één voor de lengte en de ander voor de kleur. Schrijf daarbij ook code die jouw functie aanroept, zodat je een driehoek met dezelfde kleur en lengte tekent als de code hierboven. Maak gebruik van goede naamgeving voor zowel de functie als de parameters.

## [3pt] Opgave 9:

Bekijk de code hieronder:

|  |
| --- |
| lijst = [5,7,2,3,8,1]  def mijn\_functie(lijst):    a = 0    for getal in lijst:      if getal > a:        a = getal    return a  antwoord = mijn\_functie(lijst)  print(antwoord) |

1. [1pt] Wat is de waarde van ‘antwoord’ na het uitvoeren van deze code?
2. [2pt] Vat kort samen wat de functie ‘mijn\_functie’ doet. Geef daarbij ook aan wat voor (type)invoer de functie krijgt. Let op: Er wordt gevraagd om een samenvatting, dus niet een uitleg van elke regel.

## [4pt] Opgave 10:

Schrijf een programma dat berekent hoeveel het verven van een muur kost. Doe dat in de volgende drie stappen:

1. Schrijf een functie berekenOpp dat twee getallen als **parameter** meekrijgt, namelijk de lengte en de hoogte van de muur. Oppervlakte bereken je door lengte en hoogte met elkaar te vermenigvuldigen. De functie berekent de oppervlakte levert deze op met een **return**.
2. Roep je functie aan met de getallen 3 en 5.
3. Verf kost 4 Euro per vierkante meter. Om de kosten te berekenen vermenigvuldig je het antwoord van berekenOpp met 4 en sla je dit op een in variebele. Druk de kosten vervolgens af.

## [1pt] Opgave 11:

Je wilt een gebruiker om een wachtwoord vragen. Als het wachtwoord goed is ben je klaar, anders vraag je het wachtwoord opnieuw. De gebruiker krijgt hooguit 3 pogingen. Kies uit A, B, C of D welke constructie je gebruikt, **en leg kort uit waarom**.

1. while
2. for
3. Het maakt niet uit, je kunt óf de while óf de for gebruiken.
4. Geen van beiden. Het is niet handig om hiervoor een loop te gebruiken.

Licht je antwoord kort toe.

## [4pt] Opgave 12:

Het stroomdiagram hieronder (of op de volgende pagina) laat de algoritme zien voor de functie berekenSomPositieven. Deze krijgt een lijst van getallen binnen, berekent de som van alle positieve getallen in de lijst en levert die som op.

a) Schrijf de code voor de functie berekenSomPositieven. Gebruik hiervoor het stroomdiagram. Deze moet je dus precies vertalen!

b) In je hoofdprogramma roep je de functie aan. Sla het resultaat op in een variabele.

c) Print daarna de tekst “De som is “ gevolgd door het resultaat. Gebruik hiervoor de variabele uit stap (b).

Diagram

Description automatically generated

# UITWERKINGEN

## [2pt] ANTWOORD Opgave 1:

1. 5
2. 0
3. Niet: syntax foutmelding of 0
4. oneindig

[-1 pt per fout]

## [3pt] ANTWOORD OPGAVE 2:

x = 1

while x < 3:

  print(x)

  x += 1

print("klaar")

## [2pt] ANTWOORD OPGAVE 3

a is de parameter, dat is een float, een boolean wordt opgeleverd, en een string wordt afgedrukt

## [3pt] ANTWOORD OPGAVE 4:

Meerder antwoorden mogelijk:

* H B F A B I
* B H F A B I

H totaal = 0  
B invoer = int( input (**"Voer een getal in: "**) )  
**F while not** invoer==0:  
*A* totaal = totaal + invoer  
B invoer = int( input (**"Voer een getal in: "**) )  
Iprint(totaal)

-1pt aftrek voor elke fout

## [5pt] ANTWOORD OPGAVE 5:

lijst = [3,8,7,6,4]

teller = 0

if len(lijst) == 0:

  print("Fout")

else:

  for meting in lijst:

    if (meting >= 7):

      teller += 1

  print(teller)

punten: -1 punt per regel

## [2pt] ANTWOORD OPGAVE 6:

# D

## [3pt] ANTWOORD OPGAVE 7

Print de som van alle getallen 1 t/m 6, dus (1+2+3+4+5+6=21)

|  |  |  |
| --- | --- | --- |
| n | a | b |
| 6 | 0 | 1 |
| 6 | 1 | 2 |
| 6 | 3 | 3 |
| 6 | 6 | 4 |
| 6 | 10 | 5 |
| 6 | 15 | 6 |
| 6 | 21 | 7 |

[nb. 0 pt als onduidelijk is waar de som van berekend wordt]

[1pt] correcte trace

[2pt] samenvatting / uitleg van wat er gebeurt

## [3pt] ANTWOORD OPGAVE 8

**import** turtle  
  
*#Definitie: instructies voor een driehoek  
# met een gegeven lengte en kleur***def** driehoek( lengte, kleur ):  
 turtle.pencolor( kleur )  
 turtle.forward( lengte )  
 turtle.right(120)  
 turtle.forward( lengte )  
 turtle.right(120)  
 turtle.forward (lengte )  
  
  
turtle.pendown()  
  
*#Aanroep: teken vierkant met lengte 50*driehoek( 50, **"red"** )  
  
turtle.done()

beoordeling: -1pt voor elk fout . import op juiste plek, aanroep driehoek(…), parameters (50,red), def (aanroep incl lengte en kleur), gebruik kleur en lengte in def blok, -1 punt max voor alle kleine syntax/inspiring fouten bij elkaar. -1pt voor onduidelijke naamgeving van functie/parameters, n.b. pendown mag ook in de def geplaatst zijn

## [3pt] ANTWOORD Opgave 9

a) [1pt] 8

b) [2pt] Bepaald de grootste waarde [1pt] van een gegeven lijst [1pt] van getallen.

## [4pt] ANTWOORD Opgave 10:

**def** calcArea(length, width):  
 area = length \* width  
 **return** area  
  
calculatedArea = calcArea(3,5)  
kosten = calculatedArea \* 4  
print( kosten )

a) [2pt] voor de definitie met parameters en return

b) [1pt] voor aanroep van calcArea(3,5) met juiste waarden en opslaan in een variabele: calculatedArea = calcArea(3,5)  
c) [1pt] berekening van kosten en opslaan in een variabele:

kosten = calculatedArea \* 4

## [1pt] ANTWOORD Opgave 11:

while, want je weet van tevoren niet precies hoeveel pogingen hij nodig zal hebben.

Punten: Bij verkeerde toelichting: 0 punten

## [4pt] ANTWOORD Opgave 12:

### FUNCTIEDEFINITE

def berekenSomPositieven(getallenlijst):

  som = 0 #som begint op 0

  for getal in getallenlijst:

    if getal > 0:

      som += getal #tel getal op bij de huidige som

  return(som)

### HOOFDPROGRAMMA

getallenlijst = [ 0,-2, 2, 3, -1]

berekende\_som = berekenSomPositieveGetallen(getallenlijst)

print("De som is", berekende\_som)

[1pt] def met parameter

[2pt] inhoud def met for, if, opsomming en return [-1pt per fout of ontbrekende onderdeel]

[1pt] aanroep functie met parameter en opslaan waarde in een variabele:

berekende\_som = berekenSomPositieveGetallen(getallenlijst)