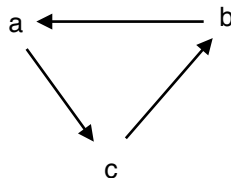


# Programmeren

0. (1 punt.) Stel, een "afhankelijk kind" is een persoon is die jonger is dan 18 jaar, en hooguit 8.000 euro verdient. Welke van de onderstaande expressies definieert een afhankelijk kind?
- a. `(leeftijd < 18) and (salaris <= 8000)`
  - b. `(leeftijd <= 18) or (salaris <= 8000)`
  - c. `(leeftijd < 18) and (salaris < 8000)`
  - d. `(leeftijd <= 18) or (salaris < 8000)`
1. (1 punt.) Wat zijn de waarden van de variabelen `girls`, `boys` en `children` nadat het volgende stukje code is uitgevoerd?
- ```
girls = 0
boys = 0
children = 0
children = girls + boys
girls = 15
boys = 12
```
- a. 0, 0, 0
  - b. 0, 0, 27
  - c. 15, 12, 0
  - d. 15, 12, 27
2. (3 punten.) In deze vraag gaan we uit van drie variabelen `a`, `b` en `c`, die een bepaalde waarde hebben (welke waarde doet niet terzake). Schrijf een stukje code om deze drie waarden één plek op te schuiven, zodat `a` de waarde krijgt van `b`, `b` de waarde krijgt van `c`, en `c` de waarde krijgt van `a`. Het volgende diagram laat zien in welke richting de waarden geschoven moeten worden:



3. (1 punt.) Wat zal de waarde van de variabele `z` zijn nadat de volgende code is uitgevoerd?

```
x = 1
y = 2
z = 3
if x < y:
    if y > 4:
        z = 5
    else:
        z = 6
```

4. (1 punt.) Beschouw het volgende blok code, waar de variabelen `a`, `b` en `c` gebruikt worden om gehele getallen (integers) op te slaan:

```
if a > b:
    if b > c:
        answer = c
    else:
        answer = b
elif a > c:
    answer = c
else:
    answer = a
```

Welke van de volgende waarden voor de variabelen `a`, `b` en `c` zullen ervoor zorgen dat bij uitvoeren van de code de variabele `answer` dezelfde waarde krijgt als variabele `b`? Kies één antwoord.

- a. `a = 1, b = 2, c = 3`
  - b. `a = 1, b = 3, c = 2`
  - c. `a = 2, b = 1, c = 3`
  - d. `a = 3, b = 2, c = 1`
5. (1 punt.) Wat zal de waarde van `result` zijn zodra het volgende stukje code is uitgevoerd?

```
nums1 = [1, -5, 2, 0, 4, 2, -3]
nums2 = [1, -5, 2, 4, 4, 2, 7]
result = 0
j = 0

while j < len(nums1):
    if nums1[j] <> nums2[j]:
        result = result + 1
    j = j + 1
```

Tip: de operator `<>` betekent hetzelfde als `!=`, namelijk "niet gelijk aan".

6. (1 punt.) Wat is de uitkomst ofwel het waarschijnlijke doel van het volgende stukje code?

```
result = 0
for j in range(0, len(number)):
    if number[j] < 0:
        result = result + 1
```

- a. om het kleinste getal in de lijst te vinden
  - b. om de negatieve getallen in de lijst te tellen
  - c. om de negatieve getallen in de lijst bij elkaar op te tellen
  - d. om 1 op te tellen bij elk van de negatieve getallen in de lijst
  - e. om de index te vinden van het eerste negatieve getal in de lijst
7. (2 punten.) Wat is de uitkomst ofwel het waarschijnlijke doel van het volgende stukje code? Formuleer zelf je antwoord in de vorm van een korte zin, vergelijkbaar met de mogelijke antwoorden bij vraag 6.

```
result = 0
for count in range(1, num + 1):
    result = result + count
```

## Loterij

8. (2 punten.) Bob, Marianne, Aniek, en Marleen zitten alle vier te azen op het laatste koekje. Uiteindelijk besluiten ze het koekje onder zichzelf te verloten. Om dit eerlijk te laten verlopen willen ze een programma schrijven dat willekeurig één van de vier namen uitprint op het scherm. Na een snelle speurtocht stuiten ze op de functie `randint` van de `random` module. Door middel van de `help`-functie krijgen ze daarbij de volgende documentatie:

```
>>> import random
>>> help(random.randint)
Help on method randint in module random:

randint(self, a, b) method of random.Random instance
    Return random integer in range [a,b] including both end points.
```

Vul het onderstaande programma aan zodat na het uitvoeren hiervan één van de vier namen willekeurig wordt uitgeprint. Maak hierbij gebruik van de `randint` functie van de `random` module.

```
namen = ["Bob", "Marianne", "Aniek", "Marleen"]
```

## Lastig lezen

9. (4 punten.) De deadline kwam te snel voor Maarten en zonder aandacht te besteden aan de opmaak (style) van zijn code heeft hij het volgende ingeleverd:

```
# vraag de gebruiker om de hoogte
x = input("How high should the pyramid be? ")
while x < 0 or x >= 24:
    x = input("How high should the pyramid be? ")

#Print the pyramid
for y in range(x):
    print (" ")*((hoogte -i)- 1) + ("#" * i + "##")
```

Dit stukje code print de halve pyramide van **mario**, zoals in de opdracht van de eerste week. Schrijf hieronder dezelfde code, maar dan met een consistente en goed leesbare style.

# Bugs in de machine

10. (4 punten.) Om wisselgeld te tellen heeft Dominique het volgende programma geschreven:

```
wisselgeld = input("Hoeveel wisselgeld (in centen) krijgt de klant? ")

# tel aantal kwartjes
aantalKwartjes = 0
while wisselgeld < 25:
    wisselgeld -= 25
    aantalKwartjes += 1

# tel aantal dubbeltjes
aantalDubbeltjes = 0
while wisselgeld >= 10:
    wisselgeld -= 10
    aantalDubbeltjes + 1

# tel aantal stuivers
aantalStuivers = 0
while wisselgeld >= 5:
    wisselgeld -= '5'
    aantalStuivers += 1

# tel aantal centen
aantalCenten = wisselgeld

print "Kwartjes:", aantalKwartjes
print "Dubbeltjes:", aantalDubbeltjes
print "Stuivers:", aantalStuivers
print "Centen:", aantalCenten
```

Dit programma zou het minimaal aantal kwartjes, dubbeltjes, stuivers en centen die nodig zijn om het wisselgeld terug te geven moeten uitprinten. Maar het werkt niet helemaal. Roepen we het programma aan, en kijken we naar de output, dan klopt die niet met onze verwachting.

```
>>> python2 change.py
Hoeveel wisselgeld (in centen) krijgt de klant? 33
Kwartjes: 0
Dubbeltjes: 0
Stuivers: 0
Centen: 3

>>> python2 change.py
Hoeveel wisselgeld (in centen) krijgt de klant? 100
Kwartjes: 0
Dubbeltjes: 0
Stuivers: 0
Centen: 0

>>> python2 change.py
Hoeveel wisselgeld (in centen) krijgt de klant? 45
Traceback (most recent call last):
  File "wisselgeld.py", line 18, in <module>
    wisselgeld -= '5'
TypeError: unsupported operand type(s) for -=: 'int' and 'str'
```

Speur alle fouten in de code op, en geef voor elke fout een mogelijke oplossing.