

# Programmeren KI

Deeltentamen 2

22 december 2017

Schrijf meteen jouw naam en studentnummer hieronder:

---

Je mag de vragen in Engels of Nederlands beantwoorden.

Dit is een "gesloten boek"-tentamen. Je mag voor het invullen je pen of potlood gebruiken, maar verder niets. Schrijf duidelijk en niet te groot.

Het laatste blaadje dient als kladpapier. Deze mag je eraf scheuren. Laat het weten als je meer kladpapier nodig hebt.

Je hoeft geen comments in je code te schrijven. Je mag alle aangeleverde functies uit de lectures of de opdrachten gebruiken in je antwoorden. Pseudocode kan ook punten opleveren, maar doorgaans minder dan "echte" code!

**Schrijf jouw antwoorden op het tentamen.**

Tijd: 2 uur

/ 32 punten
-------------

## Menselijke interpreter (6 punten)

0. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
x = 10
y = x / 2
print(y)
```

- (a) 5
- (b) 5.0
- (c) x / 2
- (d) nothing, error

1. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
print('A' + 3)
```

- (a) A + 3
- (b) 68
- (c) D
- (d) nothing, error

2. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
letters = "ABC"
magic = []
for i in range(letters):
    magic.append(letters[i:])
print(magic)
```

- (a) ABCABCABC
- (b) AABABC
- (c) ABCBCC
- (d) ["AABABC"]
- (e) ["ABCBCC"]
- (f) ["A", "AB", "ABC"]
- (g) ["ABC", "BC", "C"]
- (h) nothing, error

3. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
name = "deanna"
for i in range(len(name)):
    if name[i] == "a":
        name[i] = "x"
print(name)
```

- (a) deanna
- (b) dexnnx
- (c) dexnna
- (d) nothing, error

4. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
L = [1,2,3]
L.remove(2)
print(L)
```

- (a) [1,2,3]
- (b) [1,3]
- (c) [1,2]
- (d) nothing, error

5. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
L = ["wax", "on", "wax", "off"]
magic = []
for x in L:
    for y in x:
        if y not in magic:
            magic.append(y)
print(magic)
```

- (a) ["waxonwaxoff"]
- (b) ["w", "a", "x", "o", "n", "f"]
- (c) []
- (d) ["f", "n", "o", "x", "a", "w"]
- (e) nothing, error

## Comprehend this (6 punten)

6. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
l = [1, 2, 3]
l = [x * x for x in l]
print(l)
```

- (a) [1,2,2,3,3,3]
- (b) [1,4,9]
- (c) [1,2,3,2,3,3]
- (d) nothing, error

7. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
s = "you are gonna need a bigger boat"
l = s.split(" ")
print([x for x in l if "e" in x])
```

- (a) ["you", "are", "gonna", "need", "a", "bigger", "boat"]
- (b) ["e", "e", "e", "e"]
- (c) ["are", "need", "bigger"]
- (d) nothing, error

8. (1 punt) Het volgende stukje code produceert een lijst L. Met welke list comprehension bereiken we het zelfde resultaat?

```
L = []
for i in range(5):
    if i % 2 == 0:
        L.append(i * 2)
```

- (a) L = [i % 2 == 0 for i in range(5)]
  - (b) L = [i \* 2 if i % 2 == 0 for i in range(5)]
  - (c) L = [i \* 2 if i % 2 == 0 else 0 for i in range(5)]
  - (d) L = [i \* 2 for i in range(5) if i % 2 == 0]
9. (1 punt) Schrijf een list comprehension die de lijst [0,5,10,15,20] produceert.

10. (2 punten) Schrijf een list comprehension die de wortel trekt van elk getal in L. Hier-voor is de functie `sqrt()` geïmporteerd. De functie `sqrt()` accepteert een getal als argument en geeft de wortel daarvan terug. Bijvoorbeeld `sqrt(4)` geeft 2.0.

```
from math import sqrt
L = [4, 9, 16, 25, 36]
```

## Woordenboeken (4 punten)

11. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
d = {1:3, 2:4, 3:6}
print(d[4])
```

- (a) 2
  - (b) 4
  - (c) 6
  - (d) nothing, error
12. (3 punten) Schrijf een stukje code dat de dictionary **namen** vult. De dictionary moet als keys de voornamen van de assistenten hebben, en daarbij als value de bijbehorende achternaam. De achternaam op index 0 in de lijst **achternamen** hoort bij de voornaam op index 0 in de lijst **voornamen**. **Let op:** je mag **niet** aannemen dat er precies 4 assistenten zijn. Je mag ook **niet** aannemen dat zij altijd precies onderstaande namen hebben. De code moet dus ook werken voor andere **voornamen** en **achternamen**!

```
voornamen = ["Houda", "Mike", "Mozes", "Putri"]
achternamen = ["Alberts", "Brink", "van de Kar", "van der Linden"]
namen = {}
```

## Priemgetallen (6 punten)

13. (3 punten) Maak onderstaande functie `is_prime()` af. Deze functie accepteert een `number` (integer) als argument en moet `True` returnen als `number` een priemgetal is, anders `False`. Een getal is een priemgetal als het groter of gelijk is aan 2 en niet deelbaar door een ander positief getal anders dan 1 of zichzelf.

```
def is_prime(number):
```

14. (3 punten) Maak onderstaande functie `primes_under()` af. Deze functie accepteert een `number` (`integer`) als argument en moet alle priemgetallen onder en inclusief `number` uitprinten. Een voorbeeld aanroep:

```
>>> primes_under(5)
2
3
5
```

De volgorde waarin je de priemgetallen uitprint maakt niet uit. Je mag aannemen dat de functie `is_prime()` bestaat en werkt zoals beschreven in vraag 13.

```
def primes_under(number):
```

## Whoops! (6 punten)

Het programma hieronder was bedoeld om te kijken of een lijst gesorteerd is. Het werkt echter niet helemaal.

```
is_sorted.py
1) def is_sorted(L):
2)     for i in range(len(L)):
3)         if L[i] > L[i + 1]:
4)             print("not sorted!")
5)             return
6)     print("sorted!")
7) is_sorted([2,4,6,8])
```

Bij een test van de code krijgen we de volgende error:

```
~/workspace/ $ python3 is_sorted.py
Traceback (most recent call last):
  File "is_sorted.py", line 7, in <module>
    is_sorted([2,4,6,8])
  File "is_sorted.py", line 3, in is_sorted
    if L[i] > L[i + 1]:
IndexError: list index out of range
```

15. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

Het programma hieronder was bedoeld om een twee dimensionale lijst (een lijst in een lijst) van  $n$  bij  $n$  te maken met daarin alleen maar nullen. Het werkt echter niet helemaal.

```
zeros.py
1) def zeros(n):
2)     matrix = []
3)     for i in range(n):
4)         matrix.append([])
5)         for j in range(n):
6)             matrix.append(0)
7)     return matrix
8) print(zeros(3))
```

Bij een test van de code gebeurt het volgende:

```
~/workspace/ $ python3 zeros.py
[[], 0, 0, 0, [], 0, 0, 0, [], 0, 0, 0]
```

Zoals je ziet is de uitkomst wat vreemd. De gewenste uitkomst is:

```
[[0,0,0], [0,0,0], [0,0,0]]
```

16. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.



Aangezien het daarnet niet helemaal lukte om te kijken of een lijst gesorteerd stond, nog een poging om op een recursieve manier te kijken of een lijst gesorteerd is. Het werkt echter niet helemaal.

```
is_sorted_rec.py
1) def is_sorted(L):
2)     return L[0] < L[1] and is_sorted(L[1:])
3) print(is_sorted([1,2,3,4,5]))
```

Bij een test van de code gebeurt het volgende:

```
~/workspace/ $ python3 is_sorted_rec.py
Traceback (most recent call last):
  File "is_sorted_rec.py", line 3, in <module>
    print(is_sorted([1,2,3,4,5]))
  File "is_sorted_rec.py", line 2, in is_sorted
    return L[0] < L[1] and is_sorted(L[1:])
  File "is_sorted_rec.py", line 2, in is_sorted
    return L[0] < L[1] and is_sorted(L[1:])
  File "is_sorted_rec.py", line 2, in is_sorted
    return L[0] < L[1] and is_sorted(L[1:])
[Previous line repeated 1 more times]
IndexError: list index out of range
```

**Let op:** het doel van het programma is dat de operatie recursief gebeurt. Een iteratieve oplossing telt dus niet!

17. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

## Lastig lezen en begrijpen (4 punten)

Onderstaande oplossing voor de vigenere encryption in vigenere.py is vrij lastig te lezen en te begrijpen.

```
vigenere.py
1) import sys
2)
3) # if statement to make sure the program can function as needed
4) if len(sys.argv) != 2:
5)     print("usage: python vigenere.py keyword")
6) else:
7)     # keyword is given in command line arg 1 from sys.argv
8)     keyword = sys.argv[1]
9)
10)    # user inputs the text that she wants to have encrypted
11)    it = input("plaintext: ")
12)
13)    # count
14)    count = 0
15)
16)    # printing ciphertext to the user!
17)    print("ciphertext: ", end='', sep='')
18)
19)    # a for loop
20)    for j in range(len(it)):
21)        # make string into int
22)        i = ord(it[j])
23)        # increment the text
24)        t = 0
25)        if ord(keyword[count % len(keyword)]) in range(ord("A"), ord("Z") + 1):
26)            t = ord(keyword[count % len(keyword)]) - ord("A")
27)
28)        elif ord(keyword[count % len(keyword)]) in range(ord("a"), ord("z") + 1):
29)            t = ord(keyword[count % len(keyword)]) - ord("a")
30)
31)        # add encryption to ascii
32)        # first for capital letters, second elif for lower
33)        if i in range(ord("A"), ord("Z") + 1):
34)            x = i - ord("A")
35)            x += t
36)            # make sure not too big
37)            x = x % 26 + ord("A")
38)            print(chr(x), end='')
39)            count += 1
40)        elif i in range(ord("a"), ord("z") + 1):
41)            x = i - ord("a")
42)            x += t
43)            x = x % 26 + ord("a")
44)            print(chr(x), end='')
45)            count += 1
46)        else:
47)            print(it[i], end='', sep='')
48)            # not increase count if number
```

20. (2 punten) Geef twee verschillende verbeterpunten met betrekking tot design.

21. (2 punten) Geef twee verschillende verbeterpunten met betrekking tot style.



# Kladpapier

Wij kijken niet naar wat je op dit vel schrijft!