Programmeren

Oefententamen 2

Naam:

Studentnummer:

Schrijf jouw antwoorden op dit tentamen

Tijd: 2 uur

Maximaal aantal punten: 32

Menselijke interpreter (6 punten)

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

```
x = 10
y = x / 4
print(y)
```

   (a) `2`
   (b) `2.5`
   (c) `3`
   (d) nothing, error

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

```
print('A' + 'B')
```

   (a) `A + B`
   (b) `131`
   (c) `AB`
   (d) nothing, error

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

```
letters = "ABC"
magic = ""
for letter in letters:
    magic = letter + magic + letter
print(magic)
```

   (a) `ABC`
   (b) `CBA`
   (c) `CBAABC`
   (d) `ABCCBA`
   (e) nothing, error

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

```
name = "thomas"
for letter in name:
    if letter == "o" or letter == "a":
        letter = "x"
print(name)
```

   (a) `thxmxs`
   (b) `thomas`
   (c) `thomxs`
   (d) nothing, error

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

```
l = [1,2,3]
l.append("hello")
del l[1]
print(l)
```

   (a) `[1,2,3,"hello"]`
   (b) `[2,3,"hello"]`
   (c) `[1,3,"hello"]`
   (d) nothing, error

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

```
l = []
for i in range(3):
    l.append([])
    for j in "ABC":
        l[i].append(j)
print(l)
```

   (a) `["ABC", "ABC", "ABC"]`
   (b) `[["ABC"], ["ABC"], ["ABC"]]`
   (c) `[["A","B","C"], ["A","B","C"], ["A","B","C"]]`
   (d) nothing, error

Comprehend this (5 punten)

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

```
l = [2, 4, 6]
l = [x * 3 for x in l]
print(l)
```

   (a) `[6,12,18]`
   (b) `[2,4,6,2,4,6,2,4,6]`
   (c) `[2,2,2,4,4,4,6,6,6]`
   (d) nothing, error

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

```
s = "You've got to ask yourself one question: Do I feel lucky?"
l = s.split(" ")
print([x for x in l if "e" in x])
```

   (a) `["You've", "yourself", "one", "question:", "feel"]`
   (b) `["e", "e", "e", "e", "e", "e"]`
   (c) `"You've yourself one question: feel"`
   (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(10):
    if i % 2 == 0:
        L.append(i)
```

(a) `L = [i for i in range(10)]`

(b) `L = [i if i % 2 == 0 for i in range(10)]`

(c) `L = [i if i % 2 == 0 else 0 for i in range(10)]`

(d) `L = [i for i in range(10) if i % 2 == 0]`

9. (1 punt) Het volgende stukje code produceert een lijst `out`. Met welke list comprehension bereiken we het zelfde resultaat?

```
L = [[1,2,3], [4,5,6], [7,8,9]]
out = []
for y in L:
    for x in y:
        out.append(x)
```

(a) `out = [x for y in L for x in y]`

(b) `out = [x for x in y for y in L]`

(c) `out = [x for x in y in L]`

(d) `out = [y for x in L]`

10. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
def magic(l):
    return all([len(x) == 0 for x in l])
l = [["Hello", "Bye"], ["Oh", "My!"], [""]]
print([magic(words) for words in l])
```

(a) `[False, False, True]`

(b) `[""]`

(c) `[["Hello", "Bye"], ["Oh", "My”]]!`

(d) `nothing, error`

Woordenboeken (2 punten)

12. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

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

(a) `Hello`
(b) `1`
(c) `6`
(d) nothing, error

13. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
d = {1:2, 2:3, 3:4}
k = 1
for i in range(2):
    k = d[k]
print(k)
```

(a) `1`
(b) `2`
(c) `3`
(d) `4`
(e) nothing, error

Woorden tellen (10 punten)

Voor semantische analyse willen we weten welke woorden worden gebruikt in een stuk tekst en hoe vaak deze worden gebruikt. We hebben een nieuwsartikel van nu.nl opgeslagen in een bestand genaamd `wielklem.txt`. In het bestand staan enkel woorden met eventueel hoofdletters, en punten en komma's als leestekens. Er staan geen enters (newlines: `\n`) in het bestand. Alle woorden zijn gescheiden door een spatie. Het bestand ziet er als volgt uit

```
Politiek zet vraagtekens bij invoering wielklem Utrecht Politici hebben mondelinge
vragen gesteld over mogelijke invoering van de wielklem. Vorige week werd bekend
dat Utrecht dit in de aanbesteding heeft opgenomen. Fractievoorzitter Sander
van Waveren van het CDA wil weten of het college de wielklem daadwerkelijk als
handhavingsmiddel in wil zetten. De CDAer pleit hier al enkele jaren voor. Op
dit moment loopt de gemeente ongeveer honderdvijftigduizend euro per jaar mis
omdat boetes niet geïnd kunnen worden, stelde de fractievoorzitter. Het is voor
de gemeente lastig om buitenlandse kentekens op te sporen. Verder wilde hij van
het college meer gedetailleerde informatie het gebruik van de klem, onder andere
onder welke voorwaarden deze gebruikt zal worden en of het college inzicht heeft
in wat het voor de financiën zou betekenen. Wethouder Lot van Hooijdonk (GroenLinks)
wil afwachten welke resultaten het nog lopende onderzoek naar maatregelen om
de betalingsbereidheid te verhogen laat zien. Mocht er uit het onderzoek komen
dat het wenselijk is om met de wielklem te gaan werken, dan is dat aan de raad
om te besluiten. Dat gaat niet onopgemerkt voorbij, aldus de wethouder. De resultaten
van het onderzoek worden in het vierde kwartaal verwacht.
```

14. (3 punten) Zoals je ziet staan er veel hoofdletters, punten en komma's in `wielklem.txt`. Dit maakt het woorden tellen lastig. Maak hierom de onderstaande functie `clean()` af. Deze functie accepteert een string als argument. De functie verwijdert alle punten en komma's uit de string en vervangt alle hoofdletters met kleine letters. Vervolgens `return`ed de functie de resulterende string. Het vervangen van hoofdletters met kleine letters hebben wij al voor je gedaan op de eerste regel.

```
def clean(word):
    word = word.tolower()
```

15. (3 punten) Maak de functie `read()` hieronder af. Deze functie accepteert als argument een bestandsnaam en `return`ed alle woorden uit het bestand in een lijst. De functie moet alle punten en komma's verwijderen en alle letters naar lowercase zetten. Je mag aannemen dat de functie `clean()` bestaat en werkt. Woorden zijn in het bestand gescheiden door een spatie. Het openen van het bestand hebben wij al op de eerste regel voor je gedaan.

```
def read(filename):
    with open(filename) as f:
```

16. (4 punten) Maak de functie `count()` hieronder af. Deze functie accepteert een bestandsnaam en `return`ed een dictionary. In deze dictionary staan alle woorden uit het bestand als key en als value een integer die aangeeft hoe vaak dat woord voorkomt. Bijvoorbeeld een bestand met de tekst `Hello, world world` zou de volgende dictionary opleveren: `{"hello":1, "world":2}`. Je mag aannemen dat de functies `clean()` en `read()` bestaan en werken.

```
def count(filename):
```

Whoops! (6 punten)

Het programma hieronder was bedoeld om alle elementen van de lijst op een nieuwe regel uit te printen. Het werkt echter niet helemaal.

```
printlist.py
1) l = [1,2,3,4]
2) for e in l:
3)    print(l[e])
```

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

```
>>> python pythonextra.py
2
3
4
Traceback (most recent call last):
  File "printlist.py", line 3, in <module>
    print(l[e])
IndexError: list index out of range
```

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

Het programma hieronder was bedoeld om alle waardes groter dan 10 uit de dictionary `d` te verwijderen. Het werkt echter niet helemaal.

```
filterdict.py
1) d = {1:3, 2:29, 3:12, 4:4}
2) for x in d:
3)     if x > 10:
4)         del d[x]
5) print(d)
```

Bij een test van de code gebeurt het volgende:

```
>>> python filterdict.py
{1:3, 2:29, 3:12, 4:4}
```

Zoals je ziet staan er nog steeds waardes in die groter dan 10 zijn.

18. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

Het programma hieronder was bedoeld om op een recursieve manier een lijst om te draaien. Het werkt echter niet helemaal.

reverselist.py

```
1) def reverse(l):
2)     if len(l) <= 1:
3)         return l
4)     return [l[0]] + reverse(l[1:])
5) print(reverse([1,2,3]))
```

Bij een test van de code gebeurt het volgende:

```
>>> python reverselist.py
[1,2,3]
```

Zoals je ziet is de lijst niet omgedraaid.

19. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

Lastig lezen en begrijpen (3 punten)

Onderstaande oplossing voor mario's pyramide in mario.py is vrij lastig te lezen en te begrijpen.

```
funnies = -1
j = 2

while not 0 <= funnies or (funnies - 24 >= 0):
    print("height:", end="")
    funnies = int(input())
    if 0 < funnies <= 23:
        for i in range (0, funnies):
            for flooby in range(0, (funnies - j + 1)):
                print(" ", end = "")
            print("#" * j, sep="")
            j = j + 1
```

20. (1 punt) Geef één verbeterpunt 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!