Programmeren

PYTHON OEFENTENTAMEN 1

Naam:

Studentnummer:

Schrijf jouw antwoorden op dit tentamen

Tijd: 2 uur

Maximaal aantal punten: 29

Menselijke interpreter (5 punten)

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

```
print(3 * 5 / 2)
```

   (a) `7`
   (b) `8`
   (c) `7.5`
   (d) nothing, error

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

```
print('A' * 3)
```

   (a) `A * 3`
   (b) `195`
   (c) `AAA`
   (d) nothing, error

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

```
string = "HELLO"
out = ""
for x in string:
    out = letter + out
print(out)
```

   (a) `HELLO`
   (b) `HELL`
   (c) `OLLEH`
   (d) nothing, error

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

```
x = 3
if x > 1:
    y = 5
print(x + y)
```

   (a) `8`
   (b) `3`
   (c) `7`
   (d) nothing, error

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

```
l = [1,2]
del l[1]
print(l)
```

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

Comprehend this (4 punten)

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

```
print([[i] * 2 for i in range(3))
```

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

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

```
print([x for x in "If you gotta go, go with a smile." if x in "aeiou"])
```

   (a) `['o', 'u', 'o', 'a', 'o', 'o', 'i', 'a', 'i', 'e']`
   (b) `['I', 'o', 'u', 'o', 'a', 'o', 'o', 'i', 'a', 'i', 'e']`
   (c) `["If you gotta go, go with a smile."]`
   (d) nothing, error

7. (1 punt) Het volgende stukje code produceert een lijst `out`. Met welke list comprehension bereiken we het zelfde resultaat?

```
out = []
for i in range(1, 10, 2):
    out.append(i)
```

   (a) `out = [i for i in range(10)]`
   (b) `out = [i for i in range(10) if i % 2 == 1]`
   (c) `out = [i if i % 2 == 1 for i in range(10)]`
   (d) `out = [i if i % 2 == 0 else 0 for i in range(10)]`

8. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
print([x for x in [[True, True], [False, False], [True]] if any(x)])
```

   (a) `[True, True, True]`
   (b) `[[True, True], [False, False], [True]]`
   (c) `[[True, True], [True]]`
   (d) `nothing, error`

Woordenboeken (2 punten)

9. (1 punt) Wat is de uitkomst van het volgende stukje Python 3 code?

```
d = {}
d[0] = "foo"
print(d)
```

   (a) `foo`
   (b) `{"foo"}`
   (c) `{0:"foo"}`
   (d) nothing, error

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

```
l = "hello sascha"
d = {}
for x in l:
    if x not in d:
        d[x] = 1
    else:
        d[x] += 1
print(d["a"])
```

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

Functies (9 punten)

11. (3 punten) Maak de functie `filter()` hieronder af. Deze functie accepteert als argument een lijst van strings genaamd `words` en `return`ed een nieuwe lijst bestaande enkel uit de strings uit `words` zonder hoofdletters. Bijvoorbeeld bij `words` met waarde `["Hello", "world", "FOO"]` moet de uitkomst `["world"]` zijn. Je kan de functie `isupper()` gebruiken om te kijken of een karakter een hoofdletter is. Zo geeft `"a".isupper()` de waarde `False` en `"A".isupper()` de waarde `True`.

```
def filter(words):
```

12. (3 punten) Maak de functie `weighted_average()` hieronder af. Deze functie accepteert twee argumenten. Het eerste argument `grades` is een lijst van cijfers (`float`s) tussen de `1.0` en `10.0`. Het tweede argument `weights` is een lijst met gewichten (`float`s) tussen de `0.0` en `1.0`. De functie moet het gewogen gemiddelde berekenen van alle cijfers en dit `return`en. Bijvoorbeeld, met de cijfers `5.0` en `10.0` en de gewichten `0.25` en `0.75` is het gewogen gemiddelde: `(0.25 * 5.0 + 0.75 * 10.0) / 2 = 8.75`.

```
def weighted_average(grades, weights):
```

13. (3 punten) Maak de functie `flip()` hieronder af. Deze functie accepteert een dictionary genaamd `dict` als argument. De functie moet een nieuwe dictionary `return`en met de key-value paren van `dict` omgedraaid. Bijvoorbeeld bij `dict` met waarde `{1:3, 2:4}` zou de uitkomst `{3:1, 4:2}` moeten zijn. Je mag aannemen dat alle values in `dict` uniek zijn.

```
def flip(dict):
```

Whoops! (6 punten)

Het programma hieronder was bedoeld om het `x`de letter van de string `s` uit te printen. Het werkt echter niet helemaal.

```
test.py
1) s = "hello"
2) x = 4 / 2
3) print(s[x])
```

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

```
>>> python test.py
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    print(s[x])
TypeError: string indices must be integers
```

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

Het programma hieronder was bedoeld om de som van alle waardes in de dictionary uit te rekenen. Het werkt echter niet helemaal.

```
som.py
1) d = {1:2, 3:4, 5:6}
2) y = 0
3) for x in d:
4)     y += x
5) print(y)
```

Bij een test van de code gebeurt het volgende:

```
>>> python som.py
9
```

We hadden echter het antwoord 12 verwacht.

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 alle kleine letters uit de string te verzamelen en te `return`en in een lijst. Het werkt echter niet helemaal.

```
lower.py
1) def lowercase(s):
2)     if len(s) <= 0:
3)         return []
4)     if s[0].islower():
5)         return s[0] + lowercase(s[1:])
6)     else:
7)         return "" + lowercase(s[1:])
8) print(lowercase("abc"))
```

Bij een test van de code gebeurt het volgende:

```
>>> python lower.py
Traceback (most recent call last):
  File "lower.py", line 7, in <module>
    print(lowercase("abc"))
  File "lower.py", line 7, in lowercase
    return s[0] + lowercase(s[1:])
  File "lower.py", line 7, in lowercase
    return s[0] + lowercase(s[1:])
  File "lower.py", line 7, in lowercase
    return s[0] + lowercase(s[1:])
TypeError: must be str, not list
```

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

Lastig lezen en begrijpen (3 punten)

Onderstaande oplossing voor wisselgeld tellen in greedy.py is vrij lastig te lezen en te begrijpen.

```
1)  foo = 0
2)  n = float(input("How much change is owed: "))
3)  c = int(round(n * 100))
4)  while n < 0.0:
5)      n = float(input("How much change is owed: "))
6)      c = int(round(n * 100))
7)  for i in range(100):
8)      while c >= 25:
9)          c-=25
10)         foo+=1
11)     while c > 9:
12)         c -= 10
13)         foo += 1
14)     foo+=c//5
15)     c = c % 5
16)     while c > 0:
17)         c -= 1
18)         foo+=1
19) print(foo)
```

20. (1 punt) Geef één verbeterpunt met betrekking tot design.

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