

# Programmeren PYTHON OEFENTENTAMEN 2 nakijk

## (32 punten)

### (13 punten) Meerkeuzevragen

0. b, 1. c, 2. c, 3. b, 4. c, 5. c, 6. a, 7. a, 8. d, 9. a, 10. a, 12. a, 13. c

### (10 punten) Woorden tellen

**Syntactische fouten slaan we hier grotendeels over.** Behalve als je door zo'n fout niet kan beoordelen of de gewenste kennis aanwezig is. Bijvoorbeeld: een missende `)` gewoon negeren, een loop met compleet andere opbouw ala `for (i = 0 to 9)` is fout.

**Als het werkt is het goed!** We kijken hier niet naar design ;).

14. (a) (1 punt) Idee is correct (een nieuwe string opbouwen met een loop of gebruik van `.replace()`)  
(b) (1 punt) Strings zijn immutable, wordt hier correct mee omgesprongen?  
(c) (1 punt) Code is correct (runt minus syntactische fouten, en zou de juiste uitkomsten genereren)
15. (a) (1 punt) Idee is correct (file wordt ingelezen, gesplitst in woorden, punten en komma's worden verwijderd (evt. via `clean()`), er wordt een lijst `gereturned`)  
(b) (1 punt) File wordt correct ingelezen, evt. met `.read()`, `.readlines()`, `.readline()` of een for-loop.  
(c) (1 punt) Code is correct (runt minus syntactische fouten, en zou de juiste uitkomsten genereren)
16. (a) (1 punt) Idee is correct (lees file in en verwijder de leestekens (evt met `read()`), een dictionary wordt aangemaakt, woorden worden geteld, dictionary met getelde woorden wordt `gereturned`).  
(b) (1 punt) Dictionary wordt correct gebruikt (woord als key en de teller als value).  
(c) (1 punt) Er wordt niet gevraagd naar een key die niet bestaat (eerste keer dat je een woord tegenkomt)  
(d) (1 punt) Code is correct (runt minus syntactische fouten, en zou de juiste uitkomsten genereren)

### (6 punten) Whoops!

We kijken ook hier niet naar design. Als het probleem wordt opgelost is het goed!

17. Bijvoorbeeld:

`e` wordt hier gebruikt alsof het een index is in de lijst `1`. Het is echter een element uit de lijst `1`. Regel 3 veranderen naar `print(e)` zou het probleem oplossen.

- (a) (1 punt) Correct gespotte fout + uitleg
- (b) (1 punt) Correcte oplossing

18. Bijvoorbeeld:

Op regel 3 wordt de key `x` vergeleken met de waarde 10. In plaats daarvan moet de value behorende bij `x` worden vergeleken met 10. Regel 3 veranderen naar `if d[x] > 10` zou het probleem oplossen.

- (a) (1 punt) Correct gespotte fout + uitleg

(b) (1 punt) Correcte oplossing

19. Bijvoorbeeld:

De lijst wordt precies niet omgedraaid. Effectief wordt telkens het eerste element van de lijst vooraan geplakt, gevolgd door de rest van de lijst. Dit omdraaien, dus regel 4 veranderen naar `return reverse(l[1:]) + [l[0]]` zou het probleem oplossen.

(a) (1 punt) Correct gespotte fout + uitleg

(b) (1 punt) Correcte oplossing

### (3 punten) Lastig lezen en begrijpen

17. (1 punt) Een correct verbeterpunt voor design genoemd. Hier zijn veel antwoorden goed, denk aan:

- (a) Haal de if-statement op regel 7 weg, en plaats alles daarbinnen twee indentatiesprongen naar links. Zo heb je minder nesting!
- (b) De for-loop op regel 9 is overbodig, je kan ook `*` gebruiken net zoals op regel 11.
- (c) De conditie op regel 4 is wel erg complex, op regel 7 staat een veel simpelere versie!

18. (2 punten) Per correct verbeterpunt voor style (let wel, bijv. 2x spaties rondom operatoren noemen is 1 verbeterpunt). Ook hier zijn veel antwoorden goed, denk aan:

- (a) Gebruik witregels om blokken code te scheiden
- (b) Gebruik comments
- (c) Let op spaties rondom operatoren
- (d) Gebruik duidelijkere namen (**funnies**)