

Code: BFVP15INF1	Hertentamen: Informatica I	
Datum: 24/01/2019	Tijd: 8:30h - 10:00h	School: ILST
Lokaal: D1.08/H1.122	Klas: BFV1	Duur: 90 min
Docent : WATS/HJUR		Aantal pagina's: 4 (exclusief voorblad) Aantal opgaven: 6
Hulpmiddelen:		
Overig hulpmiddelen:		
Opgaven inleveren: Nee		
Kladpapier inleveren: Nee		
Bijzonderheden: Computertentamen met gastaccount (zonder gebruik van internet) in tentamenmodus op BIN netwerk		
Naam student:		Studentnummer:
Klas:		

Instructies

Inloggen

Voor dit tentamen moet je ingelogd zijn als gast (password gast). Op de Desktop staan de volgende bestanden/links die je nodig hebt of kunt gebruiken om dit tentamen te maken:

1. een link naar HTML PyDoc
2. het script *submit_your_work* waarmee je de gemaakte opdrachten inlevert.
3. het bestand *ubiquitin.dss*
4. het bestand *villin.dss*

Opdrachten

Het tentamen bevat een aantal (deel) opdrachten.
Geef antwoord op de vragen in een Python file die je *xxxxxx.py* noemt waarbij *xxxxxx* staat voor je studentnummer.
In verband met het indienen is het beste de scripts op de Desktop te maken.

TIP: Programmeer per stap en kijk steeds wat er gebeurt. Gebruik *idle3* of *python3* in de terminal om stukjes code uit te proberen. Maak eerst de opgaven die je makkelijk kunt maken.

Indienen

Het gemaakte werk kan ingeleverd worden door het script *submit_your_work* te draaien in de Desktop directory als:

```
submit_your_work STUDENTNR xxxxxx.py
```

Tip: sla je werk op de Desktop op. Dan kun je je werk indienen zoals hierboven beschreven.

Op een scherm worden de ingeleverde files per computer getoond zodat te zien is of je je werk daadwerkelijk hebt ingeleverd.

Beoordeling

Bij elke opdracht staat vermeld hoeveel punten maximaal te krijgen zijn.
De opdrachten worden beoordeeld op de volgende punten:

1. Werkt het programma?
2. Doet het programma wat het moet doen?
3. Is de code goed (leesbaar, effectief)?

Als een stuk code niet werkt, maak er commentaar van en schrijf er in commentaar bij wat je bedoelt.

Het cijfer wordt als volgt bepaald:
$$\text{behaald aantal punten} / \text{totaal aantal punten} * 9 + 1$$

LEES DE OPGAVEN GOED DOOR!

Hertentamen Informatica 1

Studiejaar: 2018-2019

In alle gevallen wordt gevraagd om met Python functies en methoden de gegeven variabelen te veranderen.

Een voorbeeld:

Als gevraagd wordt om aan een lijst een extra item toe te voegen dan is het volgende **niet** toegestaan:

```
my_list = ['Adenine']  
my_list = ['Adenine', 'Thymine']
```

Het is de bedoeling dat je een lijst methode gebruikt of Python syntax gebruikt om de lijst aan te vullen:

```
my_list = ['Adenine']  
my_list.append('Thymine') # Met een methode of ...  
my_list += ['Thymine'] # ... door middel van syntax
```

Vraag 1 [2p]

Maak een variabele **voornaam** en een variabele **achternaam** aan. Geef deze variabelen de juiste waarden (je voornaam en achternaam).

Print je voornaam gevolgd door je achternaam door gebruik te maken van de variabelen met behulp van de print functie.

Vraag 2

Vraag 2a [2p]

Python heeft een built-in functie om hulp te krijgen.

Print met deze built-in functie hulp voor een lijst methode waarmee items in een lijst in omgekeerde volgorde gezet worden.

Vraag 2b [2p]

Python beschikt over een wiskunde module.

Gebruik deze module om de logaritme van **14348907** met grondtal (basis) **3** te berekenen en print het resultaat naar het scherm.

Vraag 3

Je hebt een dictionary met basen:

```
dna_bases = {'A': 'Adenine', 'T': 'Thymine', 'U': 'Uracil', 'C':  
'Cytosine'}
```

Vraag 3a [2p]

Voeg de nucleobase **Guanine** (G) toe aan de dictionary met behulp van Python syntax of een dictionary methode.

Als je hier niet uitkomt gebruik je voor de volgende opdracht:

```
dna_bases = {'A': 'Adenine', 'C': 'Cytosine', 'G': 'Guanine', 'T':  
'Thymine', 'U': 'Uracil'}
```

Vraag 3b [2p]

Verwijder nu het sleutel-waarde paar **'U': 'Uracil'**

Als je hier niet uitkomt gebruik je voor de volgende opdracht:

```
dna_bases = {'A': 'Adenine', 'C': 'Cytosine', 'G': 'Guanine', 'T':  
'Thymine'}
```

Vraag 3c [2p]

Gebruik Python syntax om de waarde van sleutel **G** uit bovenstaande dictionary naar het scherm te printen.

Vraag 3d [3p]

Het nadeel van bovenstaande methode is dat het opvragen van een waarde met een niet bestaande sleutel een (key)error geeft.

Gebruik een dictionary methode om de waarde van **G** met een dictionary methode uit bovenstaande dictionary naar het scherm te printen.

Test daarbij of de sleutel **Q** geen foutmelding geeft.

Vraag 3e [6p]

Hieronder is een DNA molecuul als string gegeven:

```
dna = "g a t c a a c c g g t t"
```

Zet het DNA molecuul om in een string met hoofdletters zonder spaties er tussen.

koppel het aan de variabele `dna_up`.

Print vervolgens `dna_up` naar het scherm.

Hierbij is het de bedoeling dat je GEEN for loop gebruikt (en geen list comprehension).

Wel mag je de data omzetten naar andere data typen en gebruik maken van de data object methoden.

Vraag 4 [10p]

Hieronder is de aminozuursequentie gegeven van het groen fluorescente eiwit:

```
gfp =  
""MSRVSKGEELFTGVVPILVELDGDVNGHKFSVSGEGEGDATYGKLTCLKFICTTGKLPVPWPTLVTTLTYG  
VQCFSRYPDHMKQHDFFKSAMPEGYVQERTIFFKDDGNYKTRAEVKFEGDTLVNRIELKGIDFKEDGNIL  
GHKLEYNYNSHNVYIMADKQKNGIKVNFKIRHNIEDGSVQLADHYQNTPIGDGPVLLPDNHYLSTQSAL  
SKDPNEKRDHMLLEFVTAAGITLGMDELYK""
```

Alle aminozuren zijn hieronder gegeven:

```
aa = 'ACDEFGHIKLMNPQRSTVWY'
```

Genereer een histogram voor het voorkomen van elk van de aminozuren.

Doe dit op een efficiënte manier.

Gebruik asterisks (*) voor het weergeven van het aantal aminozuren.

Bijvoorbeeld: De A komt 8 maal voor en hiervoor print je dus 8 maal de asterisk.

Zie hieronder voorbeeld output:

```
A *****  
C **  
D *****  
E *****  
etc...
```

Vraag 5 [15p]

mRNA wordt steeds aangemaakt en afgebroken. De aanmaak is constant, maar de afbraak betreft een percentage van de hoeveelheid die er is. Meestal is er een balans tussen aanmaak en afbraak en is er sprake van een evenwicht. Als de aanmaak en afbraaksnelheid bekend zijn, kun je berekenen hoe dat evenwicht bereikt wordt, door middel van de volgende formule:

$$\text{mRNA} += \text{aanmaak} - \text{mRNA} * \text{afbraak}$$

Gebruik 0.2 voor de aanmaak en 0.1 voor de afbraak, begin met mRNA = 0 fmol/liter.

Schrijf python code die:

- de hoeveelheid mRNA berekent totdat de verandering (aanmaak – mRNA * afbraak) kleiner is dan 0.00001
- op elke stap de hoeveelheid mRNA print
- de hoeveelheid mRNA print die in evenwicht aanwezig is (in fmol/liter)

Vraag 6 [15p]

Schrijf een script dat twee bestanden met secundaire structuur inleest:

- ubiquitine.dss
- villin.dss

De secundaire structuur is gegeven als H (helix), S (sheet) en L (loop).

Voor elke aminozuurpositie in de file is aangegeven of het aminozuur behoort tot H, S of L.

Schrijf een functie die

1. een bestand leest
2. het aantal letters (aminozuren) print
3. de percentages H, S en L teruggeeft.

Gebruik de functie voor beide bestanden.

Print welk bestand het hoogste percentage H heeft.

Print welk bestand het hoogste percentage S heeft.

Indien het je niet lukt het bestand te openen dan kun je de volgende data gebruiken:

```
ubiquitine = "SSSSSSSLLSSSSSLLLLHHHHHHHHHHHHHHLLLLSSSSSHHHSSSSSS"  
villin = "SSSHHHHLLLSSSHHHHHLLHHHHHHHHHHLLLL"
```

EINDE