



# Practicum vrijdag



Collapse context



Your answer passed the tests! Your score is 100.0%. [Submission #67018b3f88eb7d851efb3f4d (2024-10-05 18:53:51)]



## Question 1: Middelste lijst

✓ Perfect



Schrijf een programma om enkel het middelste deel van een lijst over te houden (het eerste en laatste element zitten niet meer in het resultaat). Je mag er vanuit gaan dat de lijst minstens 2 elementen bevat. Er is een variabele `l` die je moet gebruiken. Dien die niet mee in.

```
1 resultaat = []
2
3 # Jouw code komt hier (pas enkel lijst resultaat aan)
4 resultaat = l
5 resultaat.pop(0)
6 resultaat.pop()
7 # en stopt hier
8 print(resultaat)
```

Submit

## Question 2: Fibonacci lijst

✓ Perfect



Schrijf een programma dat een lijst maakt van de eerste `n` Fibonacci getallen (met `n` een natuurlijk getal) waarbij `n` aan de gebruiker gevraagd wordt. De eerste twee Fibonacci getallen zijn 1 en 1. Elke volgende is de som van de twee vorige. Voorbeeld: voor `n=6` is het antwoord `[1,1,2,3,5,8]`.

```
1 n = int(input('lengte van Fibonacci-lijst: '))
2 fib_lijst = []
3 for i in range(n):
4     if i == 0 or i == 1: fib_lijst.append(1)
5     else: fib_lijst.append(fib_lijst[i-2]+fib_lijst[i-1])
6
7 print(fib_lijst)
```

Submit

## Question 3: Zoek het maximum

✓ Perfect



Schrijf een programma om het maximum in een lijst te zoeken zonder gebruik te maken van ingebouwde functies zoals `sort`, `min`, `max`, ... Er is een variabele `l` die je moet gebruiken. Dien die niet mee in.

```
1 # Jouw code komt hier
```

```

2 maximum = l[0]
3 for i in l:
4     if i > maximum: maximum = i
5 print(maximum)

```

Submit

#### Question 4: Doorsnede

✓ Perfect

Schrijf een programma dat de doorsnede van twee lijsten berekent zonder gebruik te maken van sets (als je niet weet wat sets zijn, houden zo ;-)). De twee lijsten bevatten geen dubbels en het resultaat mag ook geen dubbels bevatten. Tip: om te testen of een waarde in een lijst zit gebruik je de in operator: 1 in [1, 2] zal True geven. Op het einde roep je de sort op (zie vb code).

Voorbeelden: - de doorsnede van [] en [] is [] - de doorsnede van [1] en [2] is [] - de doorsnede van [1] en [1] is [1] - de doorsnede van [1, 2] en [1, 2, 3] is [1, 2]

```

1 # er zijn 2 variabelen l1 en l2 met waarden resp. l1 = [1, 2] en l2 = [5, 2, 3, 1]
2
3 doorsnede = []
4
5 # hier komt je code
6 for i in l1:
7     if i in l2 and i not in doorsnede:
8         doorsnede.append(i)
9
10 # de output
11 doorsnede.sort()
12 print(doorsnede)

```

Submit

#### Question 5: Een assenstelsel

✓ Perfect

We schrijven een nieuwe versie van Geogebra, maar dan ascii-style ;-) Teken een assenstelsel. Je vraagt aan de gebruiker 2 waarden: **dimx** en **dimy** (ga er vanuit dat die strikt positief zijn). **dimx** is de lengte van de x-as, **dimy** de lengte van de y-as. Die bepalen het aantal streepjes dat er moet komen. Voor **dimx = 30** en **dimy = 10** krijg je deze output:

```

^
|
|
|
|
|
|
|
|
|
|
+----->

```

```

1 dimx = int(input('lengte van de x-as: '))
2 dimy = int(input('lengte van de y-as: '))

```

```

3 print('^')
4
5 for i in range(dimy): print('|')
6
7 print('+', end='')
8 print('-' * dimx, end='')
9 print('>')

```

Submit

Question 6:  $x=a$

✓ Perfect

Teken een assenstelsel met de rechte  $x = a$  erop. Je mag er vanuit gaan dat  $a > 0$ . Je vraagt aan de gebruiker nu 3 waarden, resp.  $dimx$ ,  $dimy$  en  $a$ . Voor  $a=10$  krijg je dan deze output.

```

^
|
|
|
|
|
|
|
|
|
+----->

```

```

1 dimx = int(input('lengte van de x-as: '))
2 dimy = int(input('lengte van de y-as: '))
3 a = int(input('a: '))
4
5 print('^')
6 for i in range(dimy): print('|' + a * ' ' + '|')
7
8 print('+', end='')
9 print('-' * dimx, end='')
10 print('>')

```

Submit

Question 7:  $y=b$

✓ Perfect

Teken een assenstelsel met de rechte  $y = b$  erop. Je mag er vanuit gaan dat  $b > 0$ . Je vraagt aan de gebruiker nu 3 waarden, resp.  $dimx$ ,  $dimy$  en  $b$ . Voor  $b=4$  krijg je dan deze output.

```
^
|
|
|
|
|
|-----
|
|
+----->
```

```
1 dimx = int(input('lengte van de x-as: '))
2 dimy = int(input('lengte van de y-as: '))
3 b = int(input('b: '))
4
5 print('^')
6 for i in range(dimy):
7     if i == dimy - b: print('|' + '-' * dimx)
8     else: print('|')
9
10 print('+ ', end='')
11 print('-' * dimx, end='')
12 print('>')
```

Submit

#### Question 8: Volle rechthoek

✓ Perfect

✕

Teken een volle rechthoek van asterisken. Je vraagt aan de gebruiker de basis en de hoogte. Voor basis 5 en hoogte 3, krijg je dan deze output.

```
*****
*****
*****
```

```
1 basis = int(input('basis: '))
2 hoogte = int(input('hoogte: '))
3
4 for i in range(hoogte):
5     print('*' * basis)
```

Submit

#### Question 9: Lege rechthoek

✓ Perfect

✕

Teken een lege rechthoek van asterisken. Je vraagt aan de gebruiker de basis en de hoogte. Voor basis 5 en hoogte 3, krijg je dan deze output.

```
*****
*   *
*****
```

```
1 basis = int(input('basis: '))
2 hoogte = int(input('hoogte: '))
3
4 print('*' * basis)
5
6 for i in range(hoogte - 2):
7     print('*' + (' ' * (basis - 2)) + '*')
8
9 print('*' * basis)
```

Submit

### Question 10: Volle driehoek

✓ Perfect

✕

Teken een volle rechthoekige, gelijkbenige driehoek van asterisken. Je vraagt aan de gebruiker de (rechthoeks)zijde. Voor zijde 5, krijg je dan deze output.

```
*
**
***
****
*****
```

```
1 zijde = int(input('zijde: '))
2
3 for i in range(zijde):
4     print('*' * (i+1))
```

Submit

### Question 11: Lege driehoek

✓ Perfect

✕

Teken een lege rechthoekige, gelijkbenige driehoek van asterisken. Je vraagt aan de gebruiker de (rechthoeks)zijde. Voor zijde 5, krijg je dan deze output.

```
*
**
* *
* *
*****
```

```
1 zijde = int(input('zijde: '))
2
3 for i in range(zijde):
4     if i in [0, 1, zijde - 1]: print('*' * (i + 1))
```

```
5     else:
6         print('*' + (' ' * (i - 1)) + '*')
```

Submit

Question 12: Samenvoegen met sort.

✓ Perfect

✕

Vertrek van 2 gesorteerde lijsten en voeg die samen zodat je opnieuw een gesorteerde lijst bekomt door gebruik te maken van sort.

```
>>> l1 = [1, 3, 6, 8]
>>> l2 = [2, 6, 9, 10]
het resultaat is dan
[1, 2, 3, 6, 6, 8, 9, 10]
```

Maak in je code gebruik van de variabelen `l1` en `l2`, maar dien die NIET mee in. Bij de extra testen zal het anders mislopen.

```
1 l = []
2
3 # Jouw code komt hier, pas enkel de lijst l aan
4 #l.extend(l1)
5 #l.extend(l2)
6 l = l1 + l2
7 l.sort()
8 # en stopt hier
9 print(l)
```

Submit

Question 13: Samenvoegen zonder sort.

✓ Perfect

✕

Doe nu hetzelfde als in de vorige vraag, maar maak geen gebruik van sort. Implementeer ook geen sorteeralgoritme. Je voegt de lijsten element per element samen door te kijken naar het kleinste element (dat dus op de eerste plaats staat) en zo stelselmatig verder te gaan in beide lijsten tegelijkertijd.

```
1 l = []
2 kleinste_element = 0
3 # Jouw code komt hier (pas enkel lijst l aan)
4 i, j = 0, 0
5 e_l1, e_l2 = list(l1), list(l2)
6
7 while i < len(l1) and j < len(l2):
8     if l1[i] < l2[j]:
9         l.append(l1[i])
10        e_l1.remove(e_l1[0])
11        i+=1
12    else:
13        l.append(l2[j])
14        e_l2.remove(e_l2[0])
15        j+=1
16
17 l.extend(e_l1)
18 l.extend(e_l2)
19
```

```
20
21 # en stopt hier
22 print(l)
```

Submit

Question 14: Gemiddelde score.

Bereken de gemiddelde score van een puntenlijst en print het resultaat, afgerond op 2 cijfers na de komma, door gebruik te maken van de ingebouwde Python functies `sum`, `round` en `len`.

Gebruik de `punten` lijst, maar definieer die niet opnieuw.

Bijvoorbeeld:

```
>>> punten = [81.23, 72.44, 52.89, 64.13, 68.74]
```

Het resultaat is dan

67.89

1

Submit