



Practicum woensdag

Collapse context

Your answer passed the tests! Your score is 100.0%. [Submission #670698ba90907f632d51dfd9 (2024-10-09 14:52:42)]



Question 1: Integer deling

✓ Perfect



Schrijf een functie `integer_division(x,a)` die een integer deling uitvoert. `x` is een positieve integer en `a` een positieve integer verschillend van 0. Maak geen gebruik van de built-in operator `//` of de conversie naar integer (`int()`). We gaan het implementeren door telkens van het deeltal de deler af te trekken tot dat niet meer kan. Pas onderstaande code zodat die correct werkt door de bugs eruit te halen met de debugger.

```
1 def integer_division(deeltal, deler):
2     """
3     doe de gehele deling van deeltal door deler
4     """
5     quotient = 0
6     while deeltal >= deler:
7         quotient += 1
8         deeltal -= deler
9
10    return quotient
11
12 print(integer_division(7, 3))
13 print(integer_division(123, 5))
14 print(integer_division(0, 7))
15 print(integer_division(7, 7))
16 print(integer_division(15, 7))
```

Submit

Question 2: Oppervlakte van een cirkel

✓ Perfect



Schrijf een functie `area_of_circle(r)` die de oppervlakte van een cirkel met straal `r` teruggeeft. (Hint: gebruik `from math import pi` om de waarde `pi` te gebruiken)

```
1 from math import pi
2
3 def area_of_circle(r):
4     return pi * (r ** 2)
5
```

Submit

Question 3: Priemgetallen

✓ Perfect



Schrijf een functie `is_priem(x)` die voor een gegeven waarde `x` bepaalt of het een priemgetal is. Het programma geeft `True` terug indien dit zo is, anders `False`. Je doet dat door na te kijken of `x` deelbaar is door 2, 3, ..., Het volstaat om dat te doen tot je aan wortel `x` zit (gebruik `math.sqrt`)

```
1 import math
2
3 def is_priem(x):
4     delers = 0
5     for getal in range(1, x + 1):
6         if x % getal == 0: delers+=1
7
8     return delers == 2
```

Submit

Question 4: Het volgende priemgetal

✓ Perfect



Schrijf een functie `volgend_priemgetal(x)`, dat een natuurlijk getal `x` als input krijgt en het eerstvolgende priemgetal berekent dat groter is dan `x`. Hergebruik je functie uit de vorige vraag, maar definieer die niet opnieuw. Je code uit de vorige opgave werd automatisch toegevoegd.

```
1 def volgend_priemgetal(x):
2     resultaat = x
3     while True:
4         if is_priem(resultaat + 1):
5             return resultaat + 1
6         else:
7             resultaat += 1
```

Submit

Question 5: Graden naar radialen

✓ Perfect



Schrijf een functie `degToRad` met als parameter een reëel getal `x`, die het getal `x` in radialen teruggeeft.

```
1 from math import pi
2 def degToRad(x):
3     return (x / 180) * pi
```

Submit

Question 6: Priemfactoren

✓ Perfect



Schrijf een functie `priemfactoren` om een getal te splitsen in priemfactoren. Geef een (gesorteerde) lijst terug met alle priemfactoren. Voor 25 krijg je zo `[5,5]`, en voor 100 `[2,2,5,5]`. Doe dit door te blijven delen door 2 tot dit niet meer kan. Ga dan verder met 3, dan 4 (wat niet zal lukken aangezien je 2 al gepasseerd bent, maar dat spaart een priemtest).

```
1 def priemfactoren (x):
2     resultaat = []
3     deler = 2
4     while deler <= x:
5         if x % deler == 0:
6             resultaat.append(deler)
7             x /= deler
8         else: deler += 1
9     return resultaat
```

Submit

Question 7: Gemiddelde

✓ Perfect



De code hieronder berekent het gemiddelde van een lijst getallen. Zoek de fouten met de debugger.

```
1 def gemiddelde_lijt(lijt):
2     if len(lijt) == 0: return None
3
4     totaal = 0
5     for i in range(len(lijt)):
6         totaal += lijt[i]
7
8     return totaal / len(lijt)
```

Submit

Question 8: Reverse String

✓ Perfect



Schrijf een functie `reverse` die een input string omkeert en als returnwaarde teruggeeft. `happy` wordt dan `yppah`. Gebruik hiervoor geen ingebouwde Python functies!

```
1 def reverse(s):
2     resultaat = ''
3     for x in s:
4         resultaat = x + resultaat
5     return resultaat
```

Submit

Question 9: Palindromen

✓ Perfect

✕

Schrijf een functie `is_palindrome` die nagaat of een gegeven woord een palindroom is of niet. De functie geeft `True` of `False` terug. Hergebruik je functie uit de vorige vraag, maar definieer die niet opnieuw. Je code uit de vorige opgave werd automatisch toegevoegd.

```
1 def is_palindrome(txt):  
2     return reverse(txt) == txt
```

Submit

Question 10: Verschil

✓ Perfect

✕

Schrijf een functie `difference` die het verschil berekent van twee lijsten. Het verschil van `[7,2,1]` en `[3,2]` is `[1,7]`. Het eindresultaat is dus een gesorteerde versie van het resultaat. Je mag ervan uitgaan dat de lijsten geen duplicaten bevatten. Gebruik geen set operaties (en converteer dus ook niet tussen een lijst en een set).

```
1 def difference(x, y):  
2     resultaat = []  
3  
4     for i in x:  
5         if i not in y:  
6             resultaat.append(i)  
7  
8     return resultaat
```

Submit