Winfried Auzinger
Kevin Sturm

Sommersemester 2019
08.05.2019

Übungsaufgaben zur VU Computermathematik

Serie 6

Test your code with examples.

Exercise 6.1: ***Breaking down integers.***

(a) Write a function int_break(n) that takes an integer n and returns a list l where the entry $l[k]$ contains the (k+1)th digit of n.

(b) Write a second function counter(n), which takes the list generated by int_break(n) and reverses its order. You are not allowed to use the list function reverse().

Hint: Use the math function floor.

Exercise 6.2: ***Counting odd and even numbers.***

(a) Write two functions *odd* and *even* that get as input an integer n and as output the number of odd and even digits, respectively.

(b) Extend the functions of item (a) to floats and count the number of odd and even fractional digits (Nachkommastellen). It is sufficient to count e.g. 10 digits.

(c) Write a function, which takes an integer n and counts the number of digits that are a prime number.

Exercise 6.3: ***Sequences. I***

Let N be an arbitrary natural number. Then we define $a_0 := N$, $a_{n+1} := a_n/2$ if a_n is even, $a_{n+1} := 3a_n + 1$ if a_n is odd, for all $n \geq 0$.

(a) Write a code that implements the sequence (a_n).

(b) Verify (numerically) that for arbitrary initial value N, the sequence ends up in the cycle $4 \to 2 \to 1 \to 4$.

Exercise 6.4: ***Sequences. II***

(a) Write a code that numerically tests Fermat's last problem. Fermat's last problem states that for integers $n > 2$, $a, b, c \geq 1$ the equation $a^n + b^n = c^n$ does not hold.

(b) Write a function that implements the sequences $x_{n+1} := \frac{2x_n^3}{3x_n^2-1}$ and $y_{n+1} := \frac{1}{2}\left(y_n + \frac{1}{y_n}\right)$. Choose an initial value > 2. What is the exact limit of both sequences?

Exercise 6.5: ***Print.***

(a) Write a function which takes two strings $s1$ and $s2$ and joins them together and then prints them.

(b) Write a function which takes a string flips the first and last word of the sentence. Don't forget the capitalization.

(c) Write a function which produces the following output

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Exercise 6.6: ***Vector product, tensor product.***

The tensor product between two number $a \in \mathbb{R}^{n_1} = \mathbb{R}^{n_1 \times 1}$ and $b \in \mathbb{R}^{n_2} = \mathbb{R}^{n_2 \times 1}$ is defined by $a \otimes b := ab^\top$.

(a) Write a function $vec\_prod(a, b)$ which takes two lists a, b of length 3 and returns the vector product $a \times b$.

(b) Write a function $tensor\_prod(a, b)$ which takes two lists a, b (not necessarily the same length) and returns the tensor product.

Exercise 6.7: ***Dictionaries.***

(a)

- Grab the string 'get me' from the following dictionary.

```
d = {'key1':[1,2,{'key2':['do not get confused',{'tough':[1,2,[['get me']]]}]}]}
```

- Grab the string 'get me' from the dictionary

```
d = {'key2':[1,[[],{'bug':{'bug':'get me'}}]]}
```

(b) Write a function which takes a dictionary and reverses all of its *keys*, but not the values. Example $\{'key1' : val1, 'key2' : val2\}$ becomes $\{'1yek' : val1, '2yek' : val2\}$.

Exercise 6.8: ***Debugging***

Go to the webpage https://docs.python.org/3.6/library/pdb.html and study the module *pdb* (p stands for python and db for debugger).

(a) What does the following code do?

```
import pdb; pdb.set_trace()
```

(b) Use (a) to set a debug point at line 6 of

```
from math import sin

print("This is a debugging test.")

def f(x):

    x = x + 1.0
    a
    return sin(x)**2 + x


print("value of f(x)", f(1))
```

Read the section *Debugger Commands* on the webpage and explain what the commands "n", "s", "l" and "c" do. Now save the previous code in a file "my_debug.py" and run **python3 my_debug.py** in the command line. Use the commands n, l, s and c to navigate in debug mode.