

Einführung in die Programmierung

Prof. Dr. Franziska Matthäus
Prof. Dr. Matthias Kaschube
Dr. Karsten Tolle
Andreas Scholl



Übungsblatt

Ausgabe: 06.12.2023
Abgabe: 23.12.2023
13:00 Uhr

Rekursion mit ChatGPT

Hinweis:

- *Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.*
Für diese Abgabe: Antworten von ChatGPT müssen Sie hierbei nicht beachten!
- *Digitale Abgaben, die nicht im Format **.pdf** oder **.txt** für Texte oder **.py** für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrerer Dateien müssen diese als **.zip** zusammengefasst werden.*
- *Achten Sie darauf die Variable **__author__** in allen Quellcode Dateien (.py) korrekt zu setzen (am Anfang des Quellcodes): **__author__ = "<Matr-Nr>, <Nachname>"***
Beispiel: `__author__ = "1234567, Tolle"`
... Leerstellen vor und nach dem „=" und Leerstelle nach dem Komma beachten, sowie keine spitzen Klammern verwenden.
- *Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein. Abgaben der Dokumentation, die per Hand geschrieben und eingescannt werden, sind nur in zuvor abgesprochenen Ausnahmefällen erlaubt.*
- *Datei- und Ordnernamen sollen keine Umlaute, diakritische oder Sonderzeichen, mit Ausnahme des Unterstrichs in der Namensmitte, enthalten!*

Σ 8 Punkte

Zusatz: Achten Sie auch auf Vorgaben Ihrer Tutorin/Tutors. Hierfür auch in das Gruppenforum im Moodle-Kurs von EPR schauen!

Programmieren mit ChatGPT

2 Punkte

Für dieses Übungsblatt bitten wir Sie die Aufgaben mit Hilfe von ChatGPT zu bearbeiten. Benutzen Sie dazu bitte die kostenfreie Version auf <https://chat.openai.com/>.

Wie Sie ChatGPT benutzen, bleibt Ihnen frei überlassen. **Dabei gibt es kein Punkteabzug durch ein Plagiat oder zu wenig Eigenleistung.**

Eine Hilfestellung finden Sie hier: <https://platform.openai.com/docs/guides/prompt-engineering>

Im Anschluss an die Abgabe geben Sie bitte alle Fragen (Prompt) und Antworten (Response) mit in Moodle ab. Dazu finden Sie ein Template mit Beispiel bei der Ausgabe des Übungsblattes.

Füllen Sie im Anschluss an die Abgabe bitte noch eine kurze Umfrage in Moodle aus, um die 2 Punkte zu erhalten.

Aufgabe 1 – Rekursion

2 Punkte

Betrachten Sie folgende Code-Fragmente. Geben Sie an, was die Funktionen jeweils *zurückgeben* und wie oft sie *aufgerufen* werden. Geben Sie zusätzlich an, ob die Funktion *linear rekursiv*, *endrekursiv* oder *nicht-linear rekursiv* ist. Versuchen Sie es zuerst ohne den Rechner. Alle Fragmente sollten lauffähig sein!

a)

```
1  def f1(n, total=0):
2      if n == 0:
3          return total
4      else:
5          return f1(n // 10, total + n % 10)
6
7  y = f1(12345)
```

Rückgabewert: 15

Anzahl Aufrufe: 5

Rekursionstyp: Linear rekursiv

b)

```
1  def f2(x):
2      if len(x) == 0:
3          return []
4      return [x[-1]] + f2(x[:-1])
5
6  y = f2([9, 8, 7, 6])
```

Rückgabewert: [6, 7, 8, 9]

Anzahl Aufrufe: 4

Rekursionstyp: Nicht-linear rekursiv

c)

```
1  def f3(a, b):
2      if b == 0:
3          return 0
4      elif b > 0:
5          return a + f3(a, b - 1)
6      else:
7          return -f3(a, -b)
8
9  y = f3(4, 3)
```

Rückgabewert: 12

Anzahl Aufrufe: 3

Rekursionstyp: Linear rekursiv

d)

```
1  def f4(m, n):
2      if m == 0:
3          return n + 1
4      elif n == 0:
5          return f4(m - 1, 1)
6      else:
7          return f4(m - 1, f4(m, n - 1))
8
9  y = f4(1, 1)
```

Rückgabewert: 3

Anzahl Aufrufe: 4

Rekursionstyp: Nicht-linear rekursiv

Aufgabe 2 – Happy Strings

4 Punkte

Ein String bestehend aus Ziffern wird als „happy“ bezeichnet, wenn er so umgeordnet werden kann (oder es bereits ist), dass ein String zweimal wiederholt wird.

Hier dazu ein Beispiel:

Der String '20230322' kann zu '02320232' umgeordnet werden, was eine Wiederholung von '0232' zweimal ist.

Schreiben Sie eine Python 3.X Funktion, die für einen gegebenen String S, der nur aus Ziffern von 0-9 besteht, die **Anzahl** der Paare (l, r) bestimmt für die folgendes gilt:

- $1 \leq l \leq r \leq |S|$. (Dabei ist $|S|$ die Länge des von S)
- Der zusammenhängende Teilstring, der aus den Zeichen von l bis r in S gebildet wird, ist „happy“.

Zum Beispiel gibt es für '20230322' insgesamt 4 Paare die diese Bedingung erfüllen (1, 6), (1, 8), (2, 7) und (7, 8).

1 von 4 Punkten: Finden Sie eine rekursive Lösung für das Problem!

Denken Sie daran Ihr Programm entsprechend zu dokumentieren (insbesondere die Problemanalyse ist wichtig). Achten Sie darauf Ihren Code entsprechend mit Docstrings und Kommentaren im Code zu versehen. Wie bereits bekannt, testen Sie ihre Funktion in einer main()-Funktion, welche nur beim Start des Moduls ausgeführt wird.

Excercise

Issue: 06.12.2023
Submission: 23.12.2023
01:00 PM

Recursion with ChatGPT

Note:

- *Solutions must not be copied from the script, the internet, or other sources. These sources may only be used with proper citations, and a significant amount of individual effort must be clearly evident in the solutions.*

For this submission: You do not need to consider answers from ChatGPT!

- *Digital submissions that are not in the format **.pdf** or **.txt** for text, or **.py** for code, will not be evaluated. In the case of multiple file submissions, they must be combined into a **.zip** file.*
- *Make sure to set the variable **__author__** correctly in all source code files (.py) (at the beginning of the source code): `__author__ = "<Matr-Nr>, <Last name>"`
Example: `__author__ = "1234567, Tolle"`
... Note spaces before and after the '=' sign, a space after the comma, and do not use angle brackets.*
- *Additionally, your name must be present in every submitted **.pdf** and **.txt** file. Submissions of documentation handwritten and scanned are only allowed in exceptional cases*
- *File and folder names should not contain umlauts, diacritics, or special characters, except for the underscore in the middle of the name!*

Σ 8 Points

Note: Also, pay attention to the instructions provided by your tutor. Please check the group forum in the Moodle course for EPR as well!

Programming with ChatGPT

2 Points

For this exercise, we ask you to complete the tasks using ChatGPT via the free version available at <https://chat.openai.com/>.

How you use ChatGPT is up to you. **There will be no deduction of points for plagiarism or insufficient original work.**

You can find guidance here: <https://platform.openai.com/docs/guides/prompt-engineering>

After submission, please include all prompts and responses in Moodle. You will find a template with examples in the exercise distribution.

Following submission, please complete a brief survey in Moodle to receive the 2 points.

Task 1 – Recursion

2 Points

Consider the following code fragments. Indicate what the functions *return* and *how often they are called*. Additionally, specify whether the function is *linear recursive*, *tail-recursive*, or *multiple (non-linear) recursive*. Try it first without using the computer. All fragments should be executable!

a)

```
1  def f1(n, total=0):
2      if n == 0:
3          return total
4      else:
5          return f1(n // 10, total + n % 10)
6
7  y = f1(12345)
```

Return value: _____

Number of calls: _____

Type of recursion: _____

b)

```
1  def f2(x):
2      if len(x) == 0:
3          return []
4      return [x[-1]] + f2(x[:-1])
5
6  y = f2([9, 8, 7, 6])
```

Return value: _____

Number of calls: _____

Type of recursion: _____

c)

```
1  def f3(a, b):
2      if b == 0:
3          return 0
4      elif b > 0:
5          return a + f3(a, b - 1)
6      else:
7          return -f3(a, -b)
8
9  y = f3(4, 3)
```

Return value: _____

Number of calls: _____

Type of recursion: _____

d)

```
1  def f4(m, n):
2      if m == 0:
3          return n + 1
4      elif n == 0:
5          return f4(m - 1, 1)
6      else:
7          return f4(m - 1, f4(m, n - 1))
8
9  y = f4(1, 1)
```

Return value: _____

Number of calls: _____

Type of recursion: _____

Task 2 – Happy Strings

4 Points

A string consisting of digits is said to be „happy“ when it can be rearranged into (or already is) a repetition of some string twice.

For example:

The string 20230322 can be rearranged into 02320232, which is a repetition of 0232 twice.

Write a Python 3.X function that, for a given string S consisting of digits 0-9, finds the **number** of pairs of integers (l,r) satisfying the following conditions:

- $1 \leq l \leq r \leq |S|$. ($|S|$ is the length of S)
- The (contiguous) substring formed of the l -th through r -th characters of S is „happy“.

For example, for '20230322', there are a total of 4 pairs that meet this condition: (1, 6), (1, 8), (2, 7), and (7, 8).

1 out of 4 points: Find a recursive solution for the problem!

Remember to document your program appropriately (especially the problem analysis is important). Ensure to add docstrings and comments in your code. As previously mentioned, test your function in a `main()` function, which is executed only at the start of the module.