

Blatt 4

Vorkurs Bauingenieurwesen - Informatik - 12.04.24

Aufgabe 1

Schreibe eine Funktion `liste_test`, die keine Argumente hat und keinen Wert zurückgibt (also nur eine Funktion ist, damit es einfach zu kontrollieren ist, wann ihr Inhalt ausgeführt wird). In der Funktion sollen die folgenden Sachen passieren.

1. Speichere die Liste `[1, 2, 5, 4, 3]` als lokale Variable `x`.
2. Gebe mit mehreren `print` Statements die Zahlen 1,2,3,4,5 in dieser Reihenfolge aus. Die Zahlen müssen dabei aus der Liste `x` gelesen werden.
3. Tausche den letzten Eintrag von `x` mit dem mittleren Eintrag.
4. benutze `x` um eine neue Liste zu bauen, die den Wert `[2, 3, 4]` hat. Speichere diese neue Liste als lokale Variable `y`.
5. Ändere den ersten Eintrag in `y` zu `-2`.
6. Benutze `if` und `else` um verschiedene Ausgaben zu erzeugen, abhängig davon, ob der zweite Eintrag von `x` jetzt auch `-2` ist.
7. Schreibe `z = x` um eine weitere lokale Variable `z` zu erzeugen.
8. Setze den zweiten Eintrag von `z` auf `-20`.
9. Geben den zweiten Eintrag von `x` aus.

```
def liste_test():
```

Überlege welche Bildschirmausgaben die Funktion produzieren könnte. Rufe erst danach `liste_test()` auf um zu verifizieren ob Python sich verhält wie erwartet.

Aufgabe 2

Schreibe eine Funktion `liste_summe`, die eine Liste `l` von ganzen Zahlen übergeben bekommt und die Summe aller Einträge ausrechnet. Wenn die Liste `l` leer ist, soll `0` zurückgegeben werden. Kannst du dieses Verhalten ohne eine `if`-Bedingung erreichen? Als Beispiel soll `liste_summe([1, 2, 3, 4])` den Wert `10` zurückgeben.

```
def liste_summe(l: list[int]) -> int:
```

Aufgabe 3

Schreibe eine Funktion `tausche`, die eine Liste `l` und zwei Indices `i1` und `i2` übergeben bekommt. Der Eintrag von `l` an Position `i1` soll mit dem Eintrag an Position `i2` getauscht werden. Es soll dabei keine Kopie von `l` erzeugt werden, weswegen auch kein Wert zurückgegeben werden muss.

```
def tausche(l: list, i1: int, i2: int):
```

Als Beispiel soll der nachfolgende Code die Ausgabe `[3, 2, 1, 0]` erzeugen.

```
x = [3, 0, 1, 2]
tausche(x, 1, 3)
print(x)
```

Aufgabe 4

Wenn man in Python eine neue Liste erzeugen will, die sie selben Einträge hat wie eine bestehende Liste `x`, aber in umgedrehter Reihenfolge, schreibt man `x[::-1]`. Schreibe eine Funktion `liste_umdrehen` die nur durch Tauschen einzelner Elemente die übergebende Liste `l` umdreht.

```
def liste_umdrehen(l: list):
```

Als Beispiel soll der nachfolgende Code die Ausgabe `[3, 2, 1, 0]` erzeugen.

```
x = [0, 1, 2, 3]
liste_umdrehen(x)
print(x)
```

Aufgabe 5

Schreibe eine Funktion `sortiert` die eine Liste `l` von ganzen Zahlen übergeben bekommt und ihre Einträge der Größe nach aufsteigend sortiert. Guke dafür `len(l)` mal alle Paare von benachbarten Einträgen von `l` an und tausche die Reihenfolge eines Paares, wenn der hintere Eintrag größer als der hintere Eintrag ist. Wieso reichen `len(l)` Wiederholungen aus um die Liste garantiert zu sortieren?

```
def sortiert(l: list[int]):
```