

Probepfprüfung

Zürich, 17. Mai 2021

Lesen Sie die folgenden Punkte aufmerksam durch, bevor Sie fortfahren.

- Bitte verwenden Sie für das Lösen der Aufgaben Ihr eigenes Papier.
- Es sind keine Hilfsmittel erlaubt, nur Stift und Papier.
- Versehen Sie jedes Blatt oben mit Ihrem Namen und Ihrer Leginummer.
- Nummerieren Sie jedes Blatt Ihrer Lösung und geben Sie die Gesamtzahl der Blätter an, zum Beispiel *1 von 5, ..., 5 von 5*.
- Geben Sie klar an, welche Aufgabe Sie bearbeiten und schreiben Sie deutlich mit einem dokumentenechten Stift mit blauer oder schwarzer Tinte. Verwenden Sie kein Tipp-Ex oder ähnliches.
- Schreiben Sie auf das erste Blatt folgende Eigenständigkeitserklärung und unterschreiben Sie diese:

Ich versichere, die Prüfung selbstständig bearbeitet zu haben und dass mir bekannt ist, dass Täuschungsversuche dem Studiensekretariat gemeldet werden.

- Die Prüfung besteht aus diesem Deckblatt und 4 nummerierten Seiten mit 12 Aufgaben.
- Sie haben 60 Minuten zur Bearbeitung der Aufgaben. Der offizielle Beginn wird bekanntgegeben. Ebenfalls wird nach Ablauf der Zeit das Ende der Bearbeitungszeit bekanntgegeben. Im Anschluss müssen Sie die Prüfung mit Ihrem Smartphone scannen und per Moodle hochladen. Hierzu dürfen Sie Ihren Computer verwenden.
Der Eingang wird Ihnen bestätigt, allerdings mit einer Verzögerung.
- Es können keine inhaltlichen Fragen beantwortet werden. Falls Ihnen etwas unklar ist, notieren Sie dies mit der Bearbeitung der entsprechenden Aufgabe.

Viel Erfolg!

Für den internen Gebrauch				
Vorname	Nachname	Leginummer	Punkte	Note

Aufgabe 1

Betrachten Sie folgenden Code.

```
1 data = {0:2, 1:4, 2:5, 3:6}
2
3 print(data[1], end=" ")
4 for i in range(1, 3):
5     print(data[i])
```

Was ist die Ausgabe?

2 Punkte

Aufgabe 2

Betrachten Sie folgenden Code.

```
1 a = 5
2 b = a + 2
3
4 if not (b == b - 1) or (a > b):
5     print(a + b)
6 else:
7     print(0)
```

Was ist die Ausgabe?

2 Punkte

Aufgabe 3

Wir wollen eine Datenstruktur erstellen, um Dreiecke zu repräsentieren. Hierzu werden einfach drei Seitenlängen a , b und c gespeichert.

- Schreiben Sie eine entsprechende Klasse `Triangle`.
- Diese soll einen Konstruktor erhalten, mit dem die drei Seitenlängen gesetzt werden. Falls Zahlenwerte übergeben werden, die 0 oder negativ sind, soll eine Warnung ausgegeben werden.
- Die Klasse hat eine Methode `get_circum`, die den Umfang des Dreiecks ausgibt.

5 Punkte

Aufgabe 4

Betrachten Sie die folgende Funktion `abs_diff()`, die für zwei gegebene ganze Zahlen den Absolutwert der Differenz (also die grössere Zahl minus die kleinere Zahl) berechnen soll.

```
1 def abs_diff(x, y):
2     d = 
3     if :
4         d = -d
5     return d
```

Füllen Sie Zeilen 2 und 3 aus.

3 Punkte

Aufgabe 5

Betrachten Sie folgenden Code.

```
1 y = [3, 6, 9, 12, 15, 18, 21, 24]
2 x = [i * i for i in y if i < 11]
3 x.pop(0)
4 print(x[1])
```

Was ist die Ausgabe in Zeile 4?

3 Punkte

Aufgabe 6

Betrachten Sie folgenden Code, der für eine Liste `data` prüfen soll, ob sie aufsteigend sortiert ist, also ihre Elemente von links nach rechts grösser werden, d. h. der zweite Wert ist grösser als der erste, der dritte Wert ist grösser als der zweite etc.

```
1 data = [1, 4, 7, 10, 12, 11, 45, 78, 101]
2 ascend = True
3 for i in range(0, ):
4     if :
5         ascend = False
6 if ascend:
7     print("Liste aufsteigend")
8 else:
9     print("Liste nicht aufsteigend")
```

Füllen Sie die Zeilen 3 und 4 aus. Geben Sie ausserdem eine möglichst niedrige, korrekte Laufzeitschranke in \mathcal{O} -Notation für das Programm an. Dabei sollen Sie in $n = \text{len}(\text{data})$ messen.

3 Punkte

Aufgabe 7

Die beiden folgenden Funktionen `by_five_1()` und `by_five_2()` prüfen, ob eine gegebene Liste eine Zahl enthält, die durch 5 teilbar ist.

```
1 def by_five_1(data):
2     for i in range(0, len(data)):
3         if data[i] % 5 == 0:
4             return True
5     return False
```

```
1 def by_five_2(data):
2     found_div = False
3     for i in range(0, len(data)):
4         if data[i] % 5 == 0:
5             found_div = True
6     return found_div
```

Geben Sie Liste von 10 Zahlen an, die in `data` gespeichert sind und für die `by_five_1` höchstens halb so viele `if`-Anweisungen ausführt wie `by_five_2`, und begründen Sie, warum dies der Fall ist.

4 Punkte

Aufgabe 8

Die folgende Funktion `quad_sum()` soll rekursiv die Summe der ersten x Quadratzahlen berechnen, also für $x = 4$ beispielsweise $1 + 4 + 9 + 16 = 30$.

```
1 def quad_sum(x):
2     if x > 1:
3         
4     else:
5         
```

Füllen Sie Zeilen 3 und 5 aus.

3 Punkte

Aufgabe 9

Wir wollen auf ungerichteten, ungewichteten Graphen arbeiten, die durch Adjazenzmatrizen dargestellt sind, die als 2-dimensionale Listen kodiert sind. Die Funktion `is_connected()` soll für einen Graphen G und zwei Indizes i und j mit $i \neq j$ testen, ob i und j über einen weiteren Knoten verbunden sind.

Wenn beispielsweise eine Kante von v_i zu v_k und ausserdem eine Kante von v_k zu v_j existiert, soll `True` zurückgegeben werden und sonst `False`.

```
1 def is_connected(G, i, j):
2     for k in range(len(G)):
3         if :
4             
5         
```

Füllen Sie die Zeilen 3, 4 und 5 aus.

5 Punkte

Aufgabe 10

Betrachten Sie die folgende rekursive Funktion `surprise()`.

```
1 def surprise(data):
2     if len(data) == 1:
3         return data[0]
4     else:
5         tmp = surprise(data[1:])
6         if tmp < data[0]:
7             return tmp
8         else:
9             return data[0]
```

Beschreiben Sie in einem Satz, was diese Funktion ausgibt, wenn sie eine Liste mit n Zahlen übergeben bekommt.

4 Punkte

Aufgabe 11

Betrachten Sie folgenden Code.

```
1 y = 3
2
3 def f(x):
4     x = y * x
5     return x
6
7 x = 1
8 x += f(1)
9
10 print(f(x))
```

Was ist die Ausgabe in Zeile 10? Begründen Sie Ihre Antwort in maximal drei Sätzen.

3 Punkte

Aufgabe 12

Die folgende Funktion `split_list()` bekommt eine Liste von ganzen Zahlen als Parameter und soll die Werte dieser Liste auf zwei Listen verteilen, und zwar die negativen Zahlen in die Liste `n_data` und die positiven Zahlen und Null in die Liste `p_data`.

```
1 def split_list(data):
2     n_data = []
3     p_data = []
4     for item in :
5         if item < 0:
6             
7         else:
8             
9     return n_data, p_data
```

Füllen Sie die Zeilen 4, 6 und 8 aus.

3 Punkte