

Musterlösung

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

Lösung 1

Die Ausgabe ist

```
4 4
5
```

Lösung 2

Die Ausgabe ist

```
12
```

Lösung 3

Die Klasse kann folgendermassen umgesetzt werden:

```
1 class Triangle:
2     def __init__(self, a, b, c):
3         if a <= 0 or b <= 0 or c <=0:
4             print("Falscher Wert uebergeben")
5         self.a = a
6         self.b = b
7         self.c = c
8     def get_circum(self):
9         return self.a + self.b + self.c
10
11 tri = Triangle(10, 19, 12)
12 print(tri.get_circum())
```

Lösung 4

Die beiden Zeilen können wie folgt ausgefüllt werden:

Zeile 2: `x - y`

Zeile 3: `d < 0`

Lösung 5

Die Ausgabe in Zeile 4 ist

```
81
```

Lösung 6

Die beiden Zeilen können wie folgt ausgefüllt werden:

Zeile 3: `len(data)-1`

Zeile 4: `data[i] >= data[i+1]`

Bis auf das erste und letzte Element werden alle Elemente der Liste 2 Mal betrachtet, weswegen sich eine Laufzeit von $\mathcal{O}(n)$ ergibt.

Lösung 7

Die Funktion `by_five_1()` macht einen Early-Exit, sobald eine Zahl gefunden wird, die durch 5 teilbar ist. Die Funktion `by_five_2()` geht die Liste immer bis zum Ende durch. Ein Beispiel wäre also

```
[5, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Lösung 8

Die beiden Zeilen können wie folgt ausgefüllt werden:

```
Zeile 3: return x*x + quad_sum(x-1)
```

```
Zeile 5: return 1
```

Lösung 9

Die drei Zeilen können wie folgt ausgefüllt werden:

```
Zeile 3: G[i][k] == 1 and G[k][j] == 1
```

```
Zeile 4: return True
```

```
Zeile 5: return False
```

Lösung 10

Es wird das Minimum der n Zahlen berechnet.

Lösung 11

Die Ausgabe in Zeile 10 ist

```
12
```

Wegen der globalen Variablen y hat x in Zeile 8 den Wert 4. Dieser Wert wird in Zeile 10 dann noch einmal mit 3 multipliziert.

Lösung 12

Die beiden Zeilen können wie folgt ausgefüllt werden:

```
Zeile 4: data
```

```
Zeile 6: n_data.append(item)
```

```
Zeile 8: p_data.append(item)
```