

Labor 1 - Deadline (3te Woche)

- Alle Aufgaben sind verpflichtend.
- Alle Lösungen der Aufgaben werden in einem einzigen Repository sein.
- Jede Lösung eines Problems wird durch Klassen getrennt (-1 Pkt)
- Variablen und Methoden müssen konsistente und repräsentative Namen haben (-1 Pkt)
- Die Aufgabe wird abgegeben, indem der Link zum Git-Repository geteilt wird (den Fortschritt bei der Lösung der Aufgabe wird berücksichtigt und es wird erwartet, dass mehrere Commits vorhanden sind und nicht nur ein einziger, der die gesamte Lösung enthält. Also viele Commits erstellen!) (-10 pkt)
- Die Lösung darf keine Kompilierungsfehler enthalten. Laufzeit Warnungen werden akzeptiert sein (-10 pkt)
- Für jede Woche, in der die Aufgabe verspätet abgegeben wird, wird ein Punkt abgezogen
- Alle in den Aufgaben genannten Zahlen sind ganze Zahlen

Aufgabe 1

Die Universität hat die folgende Regeln für die Benotung:

- Jeder Student bekommt eine **Note** zwischen **0** und **100**.
- Eine Note **weniger als 40** ist eine **nicht ausreichende** Note.

Der Professor rundet die Note mit den folgenden Regeln ab:

- Wenn die **Differenz** zwischen der **Note** und dem nächsten **Vielfachen (Multipl)** von **5 weniger als 3** ist, dann wird die Note zu dem nächsten Vielfachen (Multipl) von 5 aufgerundet.
- Wenn die Note **weniger als 38** ist, wird die Note **nicht aufgerundet**.

z.B. 84 => 85

29 => 29

1. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als Rückgabewert soll die Methode ein Array mit **nicht ausreichender Note** liefern.
2. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als Rückgabewert soll die Methode den **Durchschnittswert** liefern.
3. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als Rückgabewert soll die Methode **ein Array mit die abgerundeten Noten** liefern.
4. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als Rückgabewert soll die Methode **die maximal abgerundete Note** liefern.

Beispiel:

Ein Array von Noten: [29, 37, 38, 41, 84, 67]

1. Nicht ausreichende Noten:
[29, 37, 38]
2. Durchschnittswert (Der Durchschnitt der Noten):
49.33 (gerundet auf 2 Dezimalstellen)
3. Abgerundete Noten (Die Noten nach den Aufrundungsregeln):
[29, 37, 40, 41, 85, 67]
4. Maximale abgerundete Note (Die höchste Note nach Aufrundung):
85

Aufgabe 2

Es gibt ein Array mit **n** positiven Zahlen.

1. Finden Sie die maximale Zahl.
2. Finden Sie die minimale Zahl.
3. Finden Sie die maximale Summe von n-1 Zahlen.
z.B. [4, 8, 3, 10, 17] => 4 + 8 + 10 + 17 = 39
4. Finden Sie die minimale Summe von n-1 Zahlen.
z.B. [4, 8, 3, 10, 17] => 4 + 8 + 3 + 10 = 25

Aufgabe 3

Es gibt 2 große Zahlen. Die Zahlen sind als Array dargestellt.

1. Berechnen Sie die **Summe**. Die Zahlen haben die gleiche Anzahl an Ziffern.
z.B. [1 3 0 0 0 0 0 0] + [8 7 0 0 0 0 0 0] = [1 0 0 0 0 0 0 0]
2. Berechnen Sie die **Differenz**. Die Zahlen haben die gleiche Anzahl an Ziffern.
z.B. [8 3 0 0 0 0 0 0] [5 4 0 0 0 0 0 0] = [2 9 0 0 0 0 0 0]
3. Berechnen Sie die **Multiplikation**. Die erste Zahl ist eine große Zahl, die zweite Zahl ist nur eine Ziffer.
z.B. [2 3 6 0 0 0 0 0] * 2 = [4 7 2 0 0 0 0 0]
4. Berechnen Sie die ganzzahlige **Division**. Die erste Zahl ist eine große Zahl, die zweite Zahl ist nur eine Ziffer.
z.B. [2 3 6 0 0 0 0 0] / 2 = [1 1 8 0 0 0 0 0]

Aufgabe 4

Markus will ein USB Laufwerk und eine Tastatur kaufen. Der Elektronik-Shop hat verschiedene USB Laufwerke und Tastaturen mit verschiedene Preise.

1. Schreiben Sie eine Methode, welche die **billigste** Tastatur zurückgibt.
z.B. `[40 35 70 15 45]` => 15
2. Schreiben Sie eine Methode, welche den **teuersten** Gegenstand zurückgibt.
z.B. **Tastatur** = `[15 20 10 35]`, **USB** = `[20, 15, 40 15]` => 40
3. Schreiben Sie eine Methode, welche das **teuerste** USB Laufwerk, das Markus kaufen kann, zurückgibt.
z.B. **Preise** = `[15 45 20]`, **Budget** = 30 => 20
4. Finden Sie, anhand Markus' **Budget** und der **Preislisten** für die Tastaturen und USB-Laufwerke, den maximalen Geldbetrag, der von Markus ausgegeben wird. Wenn er nicht genug für beide hat, geben Sie stattdessen **-1** zurück. Er kauft nur die zwei benötigten Gegenstände.
z.B.
b=60, tastaturen = `[40 50 60]` und USB-Laufwerke = `[8 12]` => 50 + 8 = 58
b=60, tastaturen = `[60]` und USB-Laufwerke = `[8 12]` => -1
b=60, tastaturen = `[40 60]` und USB-Laufwerke = `[8 12]` => 40 + 12 = 52