

Aufgaben

Heads-Up: Übungen zu Funktionen sind in anderen Aufgaben mit verbaut!

Input/Output

Aufgabe 1

Schreibt ein Programm, das nach eurem Namen und euer Alter fragt und anschließend den Satz "Hallo, ! Du bist Jahre alt." ausgibt.

Aufgabe 2

Schreibt ein Programm, welches das Quadrat einer eingegebenen Zahl berechnet.

Aufgabe 3

Schreibt ein Programm, welches den Umfang und den Flächeninhalt eines Rechtecks berechnet. Der Benutzer gibt jeweils die Breite und die Höhe an.

Aufgabe 4

Schreibt ein Programm, welches den Umfang und den Flächeninhalt eines Kreises berechnet. Der Benutzer gibt den Radius an.

Conditional Statements

Aufgabe 1

Schreibt ein Programm, welches anhand des eingegebenen Alters entscheidet, ob der Benutzer voll- oder minderjährig ist.

Aufgabe 2

Schreibt ein Programm, welches ausgibt, ob die eingegebene Zahl durch 2 teilbar ist.

Aufgabe 3

Schreibt ein Programm, welches nur Alice und Bob grüßt, ansonsten soll "Dich kenne ich nicht" ausgegeben werden. Der Benutzer gibt den Namen ein.

Beispiel:

Eingabe "Bob" -> "Hallo, Bob!"

Eingabe "Peter" -> "Dich kenne ich nicht!"

Aufgabe 4

Schreibt ein Programm, welches ermittelt, ob die eingegebene Zahl ein Schaltjahr ist. Die Eigenschaften für ein Schaltjahr lauten wie folgt:

1. Das Jahr ist durch 4 teilbar
2. Wenn das Jahr durch 100 teilbar ist, dann ist es KEIN Schaltjahr
3. Wenn das Jahr allerdings durch 400 teilbar ist, dann ist es doch ein Schaltjahr

Anders formuliert: Ein Jahr ist ein Schaltjahr, wenn es...

durch 4 teilbar ist und nicht durch 100 teilbar ist oder durch 400 teilbar ist.

Aufgabe 5

Erstellt ein Programm, welches das Verhalten eines einfachen Taschenrechners repliziert. Der Benutzer soll nacheinander die erste Zahl, danach den Operator (+, -, *, /) und zuletzt die zweite Zahl eingeben.

Beispiel:

- Erste Zahl: 5
- Operator: *
- Zweite Zahl: 3
- Ergebnis: 15

Aufgabe 6

Schreibt ein Programm, welches dem Nutzer nacheinander die drei Farben rot, gelb und blau auswählen lässt.

Die Auswahl erfolgt über die Eingabe von 0 (= Farbe nicht verwenden) und 1 (= Farbe zum Mischen verwenden).

Anschließend soll die resultierende Farbe ausgegeben werden.

Beispiel:

- rot: 1
- gelb: 0
- blau: 1
- Ausgabe: "Die Resultierende Farbe ist: lila"

Schleifen

Aufgabe 1

Schreibt ein Programm, welches die Zahlen von 1 bis 100 ausgibt.
(Setzt dies bitte als for-, while- und do-while-Schleife um)

Aufgabe 2

Schreibt ein Programm, welches die Summe aller Zahlen von 1 bis zu der Zahl berechnet, die der Benutzer eingegeben hat.

Für die Lösung alle Varianten (for-, while- und do-while-Schleife) implementieren!

Aufgabe 3

Schreibt ein Programm, welches die Multiplikationstabelle von 1 bis 12 ausgibt. Verwendet die Schleifen-Variante, welche euch lieber ist.

Beispiel:

1 x 1 = 1

1 x 2 = 2

...

12 x 11 = 132

12 x 12 = 144

Aufgabe 4

Schreibt ein Programm, welches durch die Zahlen von 1 bis 100 iteriert.

Wenn die Zahl durch 3 teilbar ist, soll in der Konsole "Fizz" ausgegeben werden.

Ist die Zahl durch 5 teilbar, soll "Buzz" ausgegeben werden.

Sonderfall: Wenn die Zahl durch 3 und 5 teilbar ist, soll "FizzBuzz" ausgegeben werden.

Aufgabe 5

Schreibt ein Programm, welches ermittelt, ob die vom Benutzer eingegebene Zahl eine Primzahl ist.

Hinweis: Eine Zahl ist nur dann eine Primzahl, wenn sie nur durch 1 oder durch sich selbst teilbar ist.

Aufgabe 6

Schreibt eine Funktion, welche die Fakultät einer eingegebenen Zahl (Parameter der Funktion) berechnet und zurückgibt.

Die Fakultät einer Zahl z ist folgendermaßen definiert:

$$z! = z * (z - 1) * (z - 2) * \dots * 3 * 2 * 1$$

Das heißt auf gut Deutsch:

$$6! = 6 * (6 - 1) * (6 - 2) * 3 * 2 * 1 = 720$$

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

$$3! = 3 * 2 * 1 = 6$$

$$2! = 2 * 1 = 2$$

$$1! = 1$$

Bei der Berechnung der Fakultät werden von der ausgehenden Zahl also alle Zahlen bis 1 (inklusive) miteinander multipliziert.

Benutzt für die Lösung innerhalb der Funktion einen der Schleifentypen.

Aufgabe 7

Schreibt eine Funktion, welche ermittelt, ob in einer Liste Duplikate enthalten sind oder nicht.

Beispiel:

{1, 2, 3} -> False

{1, 2, 2, 3} -> True

{1, 2, 1} -> True

Aufgabe 8

Schreibt eine Funktion, welche die Länge des längsten Wortes eines Satzes zurückgibt.

Beispiel:

"The quick brown fox jumps over the lazy dog" -> 5

"Eine Banane ist nicht immer krumm" -> 6

Aufgabe 9

Schreibt eine Funktion, welche die Summe aller Ziffern einer Zahl ermittelt.

Beispiel:

123 -> 6 ($1 + 2 + 3 = 6$)

1337 -> 14 ($1 + 3 + 3 + 7 = 14$)

Arrays

Aufgabe 1

Schreibt ein Programm, welches die Anzahl der Zeichen und Zahlen in den Eingabe des Benutzers zählt und wieder ausgibt.

Achtung: Leerzeichen sollen nicht zu der Anzahl der Zeichen gerechnet werden!

Aufgabe 2

Schreibt eine Funktion, welche das größte Element einer Liste zurückgibt.

Beispiel:

{1, 4, 14, 3, 0} -> 14

Aufgabe 3

Schreibt eine Funktion, welche alle Elemente in umgekehrter Reihenfolge ausgibt.

Beispiel:

{1, 4, 14, 3, 0} -> 0 3 14 4 1

Aufgabe 4

Schreibt eine Funktion, welche ermittelt, ob das gesuchte Element in der Liste enthalten ist.

Beispiel:

{1, 4, 14, 3, 0}

gesuchtes Element 3 -> True

gesuchtes Element 5 -> False

Aufgabe 5

Schreibt eine Funktion, welche ermittelt, ob das eingegebene Wort ein Palindrom ist.

Achtung beim Lösen: Großbuchstaben spielen hier keine Rolle!

Beispiel:

Otto -> True

Peter -> False

Weitere Übungsaufgaben

Aufgabe 1

Aufgabe 1.1

Schreibt eine Funktion, welche ein Wort mithilfe der Caesar-Verschlüsselung verschlüsselt.

(siehe <https://de.wikipedia.org/wiki/Caesar-Verschl%C3%BCsslung>)

Für die Verschlüsselung soll der Schlüssel 3 verwendet werden.

Beispiel:

Banane -> EDQDQH

Zahnarzt -> CDKQDUCW

Aufgabe 1.2

Schreibt eine Funktion, welche ganze Sätze mithilfe der Caesar-Verschlüsselung verschlüsselt.

Für die Verschlüsselung soll der Schlüssel 3 verwendet werden.

Beispiel:

The quick brown fox jumps over the lazy dog -> WKH TXLFN EURZQ IRA MXPSV RYHU
WKH ODCB GRJ

Aufgabe 1.3

Schreibt eine Funktion, welche ermittelt, mit welchem Schlüssel ein Plaintext mithilfe der Caesar-Chiffre zu einem verschlüsselten Text umgewandelt wurde.

Beispiel:

CalculateSecretKey("banane", "EDQDQH") -> 3

CalculateSecretKey("banane", "VUHUY") -> 20

Aufgabe 2

Aufgabe 2.1

Der Treibstoff für ein Modul an einer Rakete basiert auf dessen Masse.

Genauer gesagt: Um den benötigten Treibstoff für ein Modul an einer Rakete zu berechnen, nimmt man die Masse des Moduls, teilt diese durch 3, rundet ab und zieht vom Ergebnis 2 ab.

Hier ein paar Beispiele:

- Gegeben ist die Masse 12: $12 / 3 = 4 \rightarrow 4$ abgerundet bleibt 4 $\rightarrow 4 - 2 = 2$
- Gegeben ist die Masse 14: $14 / 3 = 4,6 \rightarrow 4,6$ abgerundet ist auch 4, deshalb ist hier der benötigte Treibstoff auch 2
- Gegeben ist die Masse 1969, welche 654 Einheiten Treibstoff benötigt

Um die Rakete erfolgreich starten zu können, muss das Programm die insgesamt benötigte Treibstoffmenge für jedes Modul (also: für jede "Modul-Masse") ausrechnen und zu einem Gesamtergebnis addieren, damit am Schluss die benötigte Treibstoffmenge für die gesamte Rakete berechnet wurde.

Beispiel:

Gegeben sind folgenden Modulmassen: [12, 15]

$12 / 3 = 4 \rightarrow 4$ abgerundet bleibt 4 $\rightarrow 4 - 2 = 2$

$15 / 3 = 5 \rightarrow 5$ abgerundet bleibt 5 $\rightarrow 5 - 2 = 3$

$2 + 3 = 5$

Somit ist die gesamt benötigte Treibstoffmenge für diese Rakete 5

Aufgabe:

Schreibt eine Funktion, welche genau diese Rechnungen durchführt und das Ergebnis zurückgibt.

Verwendet dabei in der Funktion eine der Schleifentypen, die ihr kennt.

Input für diese Aufgabe:

```
int[] moduleMasses = {95423, 142796, 88137, 105610, 79299, 110633,
136792, 112578, 75168, 115615, 147584, 72145, 108822, 57753,
96827, 69117, 131220, 111193, 120295, 56240, 111190, 80740,
137267, 113183, 126821, 58966, 63556, 110977, 100328, 75367,
57371, 88235, 134475, 109071, 92653, 73347, 135186, 64534, 81198,
55423, 100060, 149555, 110905, 102826, 129023, 112618, 146542,
102579, 67193};
```

Aufgabe 2.2

Bei der ersten Teilaufgabe haben wir nun also berechnet, wie viel Treibstoff jedes einzelne Modul benötigt.

Allerdings haben wir hierbei etwas Wichtiges vergessen: Treibstoff wiegt selber auch etwas.

Das heißt auf gut Deutsch:

Die Masse von Treibstoff braucht auch wieder Treibstoff, dieser Treibstoff braucht auch wiederum Treibstoff, und so weiter.

Die Rechnung, um den benötigten Treibstoff von Treibstoff zu berechnen, ist genau die Gleiche, als wenn ich den Treibstoffbedarf eines Moduls ausrechne: durch 3 dividieren, abrunden, und 2 abziehen.

Jede Masse, welche negativen Treibstoffbedarf hätte, wird betrachtet, als hätte es keinen (0) Treibstoffbedarf.

Das heißt: für jedes Modul muss zunächst die benötigte Treibstoffmenge berechnet und zum Gesamtergebnis addiert werden. Danach muss die Treibstoffmenge dieser Treibstoffmenge berechnet werden, und so weiter.

Beispiel:

- Wir haben eine Modulmasse von 14 gegeben. $14 / 3 = 4,6 \rightarrow 4,6$ abgerundet ist 4 $\rightarrow 4 - 2$ ist 2.
- Diese Treibstoffmenge von 2 braucht auch wieder Treibstoff: $2 / 3 = 0,67 \rightarrow 0,67$ abgerundet ist 0. Somit brauchen wir hier keinen weiteren Treibstoff (da $0 - 2 = -2$, was negativ ist $\rightarrow 0$)
- Wir haben eine Modulmasse von 1969, welche zunächst 654 Einheiten Treibstoff benötigt.
- Diese 654 Treibstoff-Einheiten benötigen weitere 216 Einheiten ($654 / 3 = 218 \rightarrow 218 - 2 = 216$)
- 216 braucht noch 70 Treibstoffeinheiten, welche 21 benötigen, welche 5 benötigen. 5 braucht keinen weiteren Treibstoff.
- Somit ist dann der gesamte Treibstoffbedarf für die Modulmasse 1969 966 ($654 + 216 + 70 + 21 + 5 = 966$)
- Der benötigte Treibstoff für die Modulmasse 100756 ist 50346

Beispiel:

Gegeben sind die Modulmassen: [12, 15, 100]

- 12: $12 / 3 = 4 \rightarrow 4$ abgerundet ist 4 $\rightarrow 4 - 2 = 2 \rightarrow 2 / 3 = 0,67 \rightarrow 0,67$ abgerundet ist 0 $\rightarrow 0 - 2 = -2 \rightarrow 0$, da negative Treibstoffmengen immer als 0 behandelt werden sollen. Somit ist die benötigte Treibstoffmenge für die Modulmasse 12 insgesamt 2 ($2 + 0 = 2$)
- 15: $15 / 3 = 5 \rightarrow 5 - 2 = 3 \rightarrow 3 / 3 = 1 \rightarrow 1 - 2 = -1 \rightarrow 0$, somit ist die gesamte Treibstoffmenge für die Modulmasse 15 insgesamt 3 ($3 + 0 = 3$)
- 100: $100 / 3 = 33,34 \rightarrow 33,34$ abgerundet sind 33 $\rightarrow 33 - 2 = 31 \rightarrow 31 / 3 = 10,34 \rightarrow 10,34$ abgerundet sind 10 $\rightarrow 10 - 2 = 8 \rightarrow 8 / 3 = 2,67 \rightarrow 2,67$ abgerundet sind 2 $\rightarrow 2 - 2 = 0$

Somit ist die Treibstoffmenge für die Modulmasse insgesamt 39 ($31 + 8 + 0 = 39$)

Aufgabe:

Schreibt eine Funktion, welche die insgesamte Treibstoffmenge nach diesen Kriterien berechnet und diese zurückgibt.

Benutze für die Lösung der Aufgabe den gleichen Input von der ersten Aufgabe und berechne die insgesamt benötigte Treibstoffmenge, wenn der Treibstoff auch Treibstoff braucht.

Input für diese Aufgabe:

Siehe Aufgabe 2.1