

7 Funktionen - Rekursionen - Übungen - Lösungen

A01_Add

- Rekursive Addition von zwei ganzen Zahlen. Wie beim Fingerzählen: solange 1 dazuzählen bis Zahl erreicht.
- Pseudocode:
 - $\text{Summe} = \text{Summand1} + \text{Summand2} = \text{Summand1} + (\text{Summand2}-1) + 1$
 - $\text{Summe von Summand1} + 0 = \text{Summand1}$

L:

```
#include <stdio.h>
#include <stdlib.h>

int add(int a, int b);

int main()
{
    int x = 31234;
    int y = 50000;

    printf("%d + %d = %d\n", x, y, add(x,y));
    return 0;
}

int add(int a, int b) {
    if (b <= 0) return a;           // Abbruchbedingung
    return a = add(a, b-1) + 1;
}
```

A02_Sub

- Rekursive Subtraktion von zwei ganzen Zahlen.
- Pseudocode:
 - $\text{Differenz} = \text{Minuend} - \text{Subtrahend} = \text{Minuend} - (\text{Subtrahend}-1) - 1$
 - $\text{Differenz von Minuend} - 0 = \text{Minuend}$

A03_Mult

- Rekursive Multiplikation von zwei ganzen Zahlen: Multiplikant wird Multiplikator oft miteinander addiert.
- Pseudocode:
 - $\text{Produkt} = \text{Multiplikant} * \text{Multiplikator} = \text{Multiplikant} * (\text{Multiplikator}-1) + \text{Multiplikant}$
 - $\text{Produkt von Multiplikant} * 1 = \text{Multiplikant}$

A03_Fakultaet

- Die Fakultät einer Zahl spielt in der Wahrscheinlichkeitsrechnung aber auch in Taylorreihen eine Rolle. Berechne die Fakultät einer ganzen Zahl: $n! = 1 * 2 * 3 * \dots * n$
- Pseudocode:
 - $\text{Fakultät}(n) = \text{Fakultät}(n-1) * n$
 - $\text{Fakultät}(1) = 1$

A05_Fibonacci

- In der Natur gibt es viele Wachstumsgesetze die sich nach einer bestimmten Folge berechnen lassen: $n = (n-1) + (n-2)$. Wobei gilt: $\text{Fibo}(1) = \text{Fibo}(2) = 1$
- Pseudocode:
 - $\text{Fico}(n) = (n-1) + (n-2)$

- o $\text{Fibo}(2) = \text{Fibo}(1) = 1$