Übungsaufgabe 17

Berechnung der Fibonacci-Folge

Aufgabe: die Fibonacci-Folge soll sowohl mittels der bekannten Schleifen als auch durch eine Rekursion berechnet werden, so dass ein Funktionsaufruf fibonacci(n) den Wert der n-ten Fibonacci-Zahl zurückliefert.
Bsp:

```
function fibonacci(/* Parameter */) {
   // Berechnung
}

echo fibonacci(25);
//=> 75025
```

Die Fibonacci-Folge

Die Fibonacci-Folge ist die unendliche Folge von natürlichen Zahlen, die (ursprünglich) mit zweimal der Zahl 1 beginnt oder (häufig, in moderner Schreibweise) zusätzlich mit einer führenden Zahl 0 versehen ist.[1] Im Anschluss ergibt jeweils die Summe zweier aufeinanderfolgender Zahlen die unmittelbar danach folgende Zahl:

Ganzzahl n	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**...**	**n (n >= 3)**
Berechnung	**fix**	**fix**	**0 + 1**	**1 + 1**	**1 + 2**	**2 + 3**	**3 + 5**	**5 + 8**	...	$f_{n-2} + f_{n-1}$
Fibonaccizahl f_n	**0**	**1**	**1**	**2**	**3**	**5**	**8**	**13**	...	f_n

> Die darin enthaltenen Zahlen heißen **Fibonacci-Zahlen**. Benannt ist die Folge nach `Leonardo Fibonacci`, der damit im Jahr 1202 das Wachstum einer Kaninchenpopulation beschrieb. Die Folge war aber schon in der Antike sowohl den Griechen als auch den Indern bekannt.

> Weitere Untersuchungen zeigten, dass die Fibonacci-Folge auch noch zahlreiche andere Wachstumsvorgänge der Pflanzen beschreibt. Es scheint, als sei sie eine Art Wachstumsmuster in der Natur.[3]

> Die Fibonacci-Zahlen weisen einige bemerkenswerte mathematische Besonderheiten auf:

- Aufgrund der Beziehung zur vorherigen und zur folgenden Zahl scheint Wachstum in der Natur einem Additionsgesetz zu folgen.
- Die Fibonacci-Folge steht in einem unmittelbaren Zusammenhang zum Goldenen Schnitt. Je weiter man in der Folge fortschreitet, desto mehr nähert sich der *Quotient* aufeinanderfolgender Zahlen dem `Goldenen Schnitt` (`1,618033…`) an (beispielsweise 13:8 = 1,6250; 21:13 ≈ 1,6154; 34:21 ≈ 1,6190; 55:34 ≈ 1,6176; etc.).
- Diese Annäherung ist alternierend, d. h. die Quotienten sind abwechselnd kleiner und größer als der Goldene Schnitt.
  (wikipedia)

Definition der Fibonacci-Folge

> Die Fibonacci-Folge $f_1, f_2, f_3, \ldots$ ist durch das rekursive Bildungsgesetz

> $f_n = f_{n-1} + f_{n-2}$ für $n \geq 3$

mit den Anfangswerten

> $f_1 = f_2 = 1$

definiert. Das bedeutet in Worten:

> Für die beiden ersten Zahlen wird der Wert eins vorgegeben. Jede weitere Zahl ist die Summe ihrer beiden Vorgänger in der Folge.

n	f_n	n	f_n	n	f_n	n	f_n	n	f_n
1	1	11	89	21	10946	31	1346269	41	165580141
2	1	12	144	22	17711	32	2178309	42	267914296
3	2	13	233	23	28657	33	3524578	43	433494437
4	3	14	377	24	46368	34	5702887	44	701408733
5	5	15	610	25	75025	35	9227465	45	1134903170
6	8	16	987	26	121393	36	14930352	46	1836311903
7	13	17	1597	27	196418	37	24157817	47	2971215073
8	21	18	2584	28	317811	38	39088169	48	4807526976
9	34	19	4181	29	514229	39	63245986	49	7778742049
10	55	20	6765	30	832040	40	102334155	50	12586269025

Tabelle: die ersten 50 Fibonacci-Zahlen
(wikipedia)

Aufgabe 1: Schreibe ein PHP-Programm zur Berechnung einer Fibonacci-Zahl der Fibonacci-Folge. Zum Testen lassen wir uns die Fibonacci-Zahl zur Zahl 16 berechnen (`Ergebnis = 987`).

Aufgabe 2: Verallgemeinere Dein Programm, in dem Du die Berechnung nun in einer Funktion ausführst. Dabei soll die zu berechnende Fibonacci-Zahl `n` als Argument beim Funktionsaufruf übergeben werden und die Funktion soll uns den berechneten Wert zurückgeben.

```
// z. B. Funktionsaufruf
fibonacci(16) // => 987
```

Aufgabe 3: Entwickele nun einen rekursiven Lösungsansatz zur Berechnung einer Fibonacci-Zahl der Fibonacci-Folge.

Aufgabe 4: Optimiere nun Deinen rekursiven Lösungsansatz.