

	Übungen Programmierung Dr. J. Brose, PHY C116, Tel. 32104 J.Brose@physik.tu-dresden.de	4
---	--	---

Fibonacci-Folge

1. Schreiben Sie eine *rekursive Funktion* zum Berechnen der Fibonacci-Zahlen beliebiger Ordnung! Geben Sie im Hauptprogramm die Fibonacci-Zahl einer durch den Benutzer festgelegten Ordnung n aus. Testen Sie mit $n = 10, 20, 30, 40$!

Zur Erinnerung: Die Fibonacci-Folge f_0, f_1, f_2, \dots , benannt nach Leonardo Fibonacci, ist durch das rekursive Bildungsgesetz

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

mit den Anfangswerten

$$f_0 = 0 \text{ und } f_1 = 1$$

definiert.

2. Zeitmessungen lassen sich mit der Funktion `time.time()` aus dem Modul `time` durchführen. Ermitteln Sie die Zeit für die Berechnung der Fibonacci-Zahlen der oben angegebenen Ordnungen! Was für ein Zeitverhalten beobachten Sie?
3. Was könnte die Ursache dafür sein, dass sich große Fibonacci-Zahlen nicht mehr berechnen lassen? Implementieren Sie eine Variable, die zählt, wie oft die Funktion zur Berechnung der Fibonacci-Zahlen aufgerufen wird!
4. Beheben Sie das Problem (Funktion mit einer *iterativen* Berechnung der Fibonacci-Zahlen)!
5. Berechnen Sie jetzt die ersten 50 Glieder der Fibonacci-Folge und geben Sie das Verhältnis des aktuellen Gliedes zu seinem Vorgänger im Terminalfenster aus. Modifizieren Sie die Stellenanzahl der Ausgabe so, dass Sie Unterschiede in den ausgegebenen Werten sehen.

Zusätzlich für die Übungsgruppen am Mittwoch:

6. Beteiligen Sie sich an der Primzahl-Suche!
 Mersenne-Primzahlen sind Zahlen der Form $M_p = 2^p - 1$, $p \in \mathbb{N}$. Für Mersenne-Primzahlen ab M_3 gilt, dass $S(p-1)$ durch M_p teilbar ist, wobei die Folge $S(i)$

definiert sei durch: $S(1) = 4$ und $S(i + 1) = S(i)^2 - 2$.

Schreiben Sie eine Funktion, die die Ordnung p der Mersenne-Zahl übernimmt und testet, ob die entsprechende Zahl eine Primzahl ist oder nicht. Geben Sie die Primzahlen im Dezimal- und Binärsystem aus! Testen sie z.B. $p = 11213$!