Übung 5

Rekursionen

Benötigt: -

Aufgabe 1: Einstufige Rekursion in tiefer Realisierung

- öffne neue Funktion „summe.m“
- Erzeuge eine Funktion zur Ermittlung der Summe der ersten n natürlichen Zahlen (n>=0) nach der Rekursion $summe(n) = \begin{cases} summe(n-1)+n & , \ n > 0 \\ 0 & , \ n = 0 \end{cases}$ durch eine tiefe Realisierung.
- Teste durch die Aufrufe und vergleiche die Reaktionen:

```
>> summe(0)
ans = 0
>> summe(1)
ans = 1
>> summe(2)
ans = 3
>> summe(200)
ans = 20100
>>
```

Aufgabe 2: Einstufige Rekursion in flacher Realisierung

- öffne neue Funktion „summef.m“
- Erzeuge eine Funktion zur Ermittlung der Summe der ersten n natürlichen Zahlen (n>=0) nach der Rekursion $summe(n) = \begin{cases} summe(n-1)+n & , \ n > 0 \\ 0 & , \ n = 0 \end{cases}$ durch eine flache Realisierung.
- Teste durch die Aufrufe und vergleiche die Reaktionen:

```
>> summef(0)
ans = 0
>> summef(1)
ans = 1
>> summef(2)
ans = 3
>> summef(200)
ans = 20100
>> summef(2000)
ans = 2001000
>>
```

Aufgabe 3: Die Türme von Hanoi (einfach)


Die Türme von Hanoi

- Definition: „Die Türme von Hanoi“
  Das Spiel besteht aus drei Stäben A, B und C, auf die mehrere gelochte Scheiben gelegt werden, alle verschieden groß. Zu Beginn liegen alle Scheiben auf Stab A, der Größe nach geordnet, mit der größten Scheibe unten und der kleinsten oben. Ziel des Spiels ist es, den kompletten Scheiben-Stapel von A nach C zu versetzen.
  Bei jedem Zug darf die oberste Scheibe eines beliebigen Stabes auf einen der beiden anderen Stäbe gelegt werden, vorausgesetzt, dort liegt nicht schon eine kleinere Scheibe. Folglich sind zu jedem Zeitpunkt des Spieles die Scheiben auf jedem Feld der Größe nach geordnet.
- öffne die Funktion „bewege.m“
- Implementiere den rekursiven Randoff'schen Algorithmus entprechend dem Pseudo-Code:

```
funktion bewege (Zahl i, Stab a, Stab b, Stab c) {
   falls (i>0) {
      bewege(i-1, a, c, b)
      Ausgabe: verschiebe oberste Scheibe von a nach c
      bewege(i-1, b, a, c)
   }
}
```

  Der Algorithmus besteht im Wesentlichen aus einer Funktion **bewege**, die vier Parameter besitzt. Mit **i** ist die Anzahl der zu verschiebenden Scheiben bezeichnet, mit **a** der Stab von dem verschoben werden soll, mit **b** der Stab, der als Zwischenziel dient und mit **c** der Stab, auf den die Scheiben verschoben werden sollen.
- Teste durch den Aufruf und vergleiche die Reaktion:

```
>> bewege(3,1,2,3)
verschiebe oberste Scheibe von Stab 1 nach Stab 3
verschiebe oberste Scheibe von Stab 1 nach Stab 2
verschiebe oberste Scheibe von Stab 3 nach Stab 2
verschiebe oberste Scheibe von Stab 1 nach Stab 3
verschiebe oberste Scheibe von Stab 2 nach Stab 1
verschiebe oberste Scheibe von Stab 2 nach Stab 3
verschiebe oberste Scheibe von Stab 1 nach Stab 3
>>
```

Aufgabe 4: Die Türme von Hanoi (fortgeschritten)

- öffne die Funktion „bewege2.m“
- erweitere die Funktion bewege um die Zustandsvariable X zu
  `funktion Zustand X = bewege2(Zahl i,Stab a,Stab b,Stab c,Zustand X)`
- Der Scheibenzustand X ist eine Matrix mit so vielen Zeilen, wie Scheiben vorhanden sind und 3 Spalten, die für die drei Stäbe stehen. Die Scheiben selbst werden durch ihre Größe (ganze Zahl) dargestellt, d.h. hat ein Element in X den Wert 4, so ist an dieser Stelle eine Scheibe mit der Größe 4.
- In der Funktion bewege2 ist nun an der Stelle von „Ausgabe: …..“ die Matrix X entsprechend zu verändern (*verschiebe oberste Scheibe von a nach c*) und auszugeben. Hierdurch kann der Verlauf der Umsortierung nachvollzogen werden.
- Teste durch die Aufrufe und vergleiche die Reaktion:

```
>> X = [1 0 0;
        2 0 0;
```

```
        3 0 0];
>> bewege2(3,1,2,3,X);
    1     0     0
    2     0     0
    3     0     0

    0     0     0
    2     0     0
    3     0     1

    0     0     0
    0     0     0
    3     2     1

    0     0     0
    0     1     0
    3     2     0

    0     0     0
    0     1     0
    0     2     3

    0     0     0
    0     0     0
    1     2     3

    0     0     0
    0     0     2
    1     0     3

    0     0     1
    0     0     2
    0     0     3
>>
```

Aufgabe 5: Die Türme von Hanoi (Zählung)

- öffne die Funktion „bewege3.m“
- ändere die Funktion bewege dahingegen ab, dass sie an Stelle der Ausgabe eine globale Variable COUNT um 1 erhöht.
- Teste durch die Aufrufe und vergleiche die Reaktion:
  >> global COUNT
-
- ```
  >> for i=1:6, COUNT=0; bewege3(i,1,2,3); [i COUNT], end
  ans =     1     1
  ans =     2     3
  ans =     3     7
  ans =     4    15
  ans =     5    31
  ans =     6    63
  >>
  ```
- Kann man aus den Resultaten eine Gesetzmäßigkeit erkennen ?
  ```