Aufgabenblatt 7: Rekursion

Kapitel 1: Praktikumsaufgaben

7.1.1: Rekursive Stringfunktionen ✔

Erstellen Sie rekursive Implementierungen der Funktionen

- `strlen`
- `strcmp`

aus der Runtime-Library.

Feedback

Du hast am 17.11.16, 13:19 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 4/10
Spaß: 5/10
Zeit: 10min

7.1.2: Binomialkoeffizient rekursiv ✔

Binomialkoeffizienten können über die Formel

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k} + \binom{n-1}{k-1} & \text{wenn } n > k > 0 \\ 1 & \text{wenn } n{=}k \text{ oder } k{=}0 \end{cases}$$

berechnet werden. Diese Formel ist übrigens die Konstruktionsvorschrift für das Pascalsche Dreieck.

Erstellen Sie eine rekursive Funktion zur Berechnung von Binomialkoeffizienten auf der Grundlage dieser Formel.

Vergleichen Sie die Funktionsergebnisse mit denen einer iterativen Implementierung (siehe hier).

Feedback

Du hast am 17.11.16, 13:25 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 1/10
Spaß: 4/10
Zeit: 5min

7.1.3: Rekursiv sortieren ✔

Betrachten Sie die folgende Funktion, die den Bubblesort-Algorithmus zum Sortieren eines Zahlenarrays implementiert.

```
void bubblesort( int n, int daten[])
    {
    int i, k, t;

    for( i = n-1; i > 0; i--)
        {
        for( k = 0; k < i; k++)
            {
            if( daten[k] > daten[k+1])
                {
                t = daten[k];
                daten[k] = daten[k+1];
                daten[k+1] = t;
                 }
            }
        }
    }
```

Erstellen Sie eine rekursive Implementierung dieser Funktion.

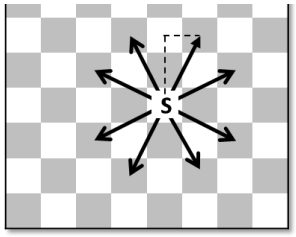
Feedback

Du hast am 17.11.16, 14:26 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 4/10
Spaß: 3/10
Zeit: 10min

7.1.4: Springer ✔

Der Springer ist eine leichte und bewegliche Figur beim Schach, die von ihrer aktuellen Position aus bis zu 8 Felder im sogenannten Rösselsprung (zwei vorwärts, eins seitwärts) mit einem Zug erreichen kann:



Erstellen Sie ein Programm, das einen Springer von einem beliebigen Startpunkt zu einem beliebigen Zielpunkt auf einem Schachbrett ziehen kann! Die vom Programm gewählte Zugfolge muss nicht optimal sein und soll bei der Ausgabe durch fortlaufende Nummern angezeigt werden:

```
Startpunkt (Zeile Spalte):1 1
Zielpunkt  (Zeile Spalte):1 2
+--+--+--+--+--+--+--+--+
| 0|47|  |  |  |41|22|  |
+--+--+--+--+--+--+--+--+
|  |  | 1|42|21|  |  |40|
+--+--+--+--+--+--+--+--+
|32|43|46|  | 2|39|20|23|
+--+--+--+--+--+--+--+--+
|45|  |31|  |19|24| 3|38|
+--+--+--+--+--+--+--+--+
|30|33|44|25| 4|37|18|  |
+--+--+--+--+--+--+--+--+
|13|26|29|34|17| 8| 5|36|
+--+--+--+--+--+--+--+--+
|28|  |14|11| 6|35|16| 9|
+--+--+--+--+--+--+--+--+
|  |12|27|  |15|10| 7|  |
+--+--+--+--+--+--+--+--+
```

Feedback

Du hast am 21.11.16, 11:09 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 5/10
Spaß: 7/10
Zeit: 25min

Kapitel 2: Vertiefung und Selbsttest

7.2.1: Zeichenkette umgekehrt ausgeben ✔

Erstellen Sie eine rekursive Funktion, um eine Zeichenkette rückwärts auszugeben.

Feedback

Du hast am 21.11.16, 13:33 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 3/10
Spaß: 1/10
Zeit: 15min
Text: char* ist nicht beschreibbar...

7.2.2: Binomialkoeffizienten rekursiv berechnen ✔

Erstellen Sie eine Funktion, die Binomialkoeffizienten nach der Formel

$$\binom{n}{k} = \begin{cases} \frac{n}{k}\binom{n-1}{k-1} & falls\ n \geq k > 0 \\ 1 & falls\ n \geq k = 0 \end{cases}$$

rekursiv berechnet. Achten Sie dabei darauf, dass keine Rechenfehler durch die Integerdivision entstehen. Vergleichen Sie das Ergebnis mit einer iterativen Implementierung (siehe hier).

Feedback

Du hast am 21.11.16, 13:40 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 3/10
Spaß: 3/10
Zeit: 5min

7.2.3: Zahlen rekursiv anordnen ✔

Erstellen Sie eine rekursive Funktion, die die Zahlen in einem Array so anordnet, dass alle negativen vor allen nicht negativen Zahlen stehen.

Folgen Sie bei der Implementierung der Idee des hier beschriebenen iterativen Verfahrens.

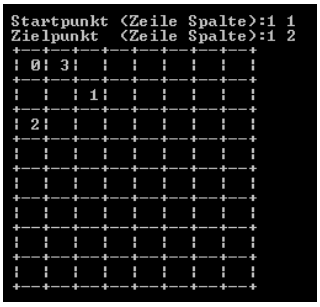
Feedback

Du hast am 21.11.16, 14:17 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett
Schwierigkeit: 3/10
Spaß: 4/10
Zeit: 10min

7.2.4: Optimale Zugfolge des Springers ✔

Erweitern Sie das Programm "Springer" so, dass eine optimale, d. h. möglichst kurze Zugfolge ermittelt wird.



Die Losung dieser Aufgabe setzt voraus, dass Sie zuvor die Aufgabe "Springer" gelöst haben. Lösen Sie also zunächst diese Aufgabe, falls es noch nicht geschehen ist.

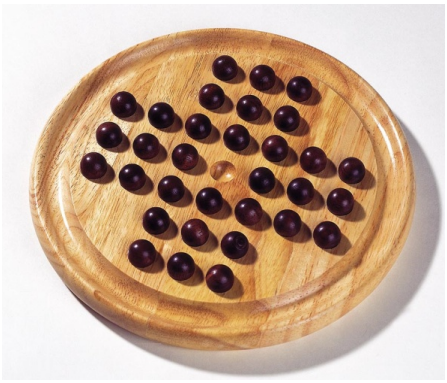
Feedback

Du hast am 21.11.16, 12:41 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: teilweise
Schwierigkeit: 6/10
Spaß: 7/10
Zeit: 25min
Text: Ansatz klar, aber die Anforderung ist nicht mit meiner Struktur aus der 1. Teilaufgabe zu vereinen. Problem ist nicht den kürzesten Weg zu finden, sondern währenddessen die Markierungen zu setzten... Spuren lassen sich nicht so einfach löschen.

7.2.5: Solitär ✔

Sie kennen das Brettspiel Solitär:



Aufgabe des Spiels ist es, alle Kugeln bis auf eine durch "Überspringen" vom Brett zu entfernen. Übersprungen werden kann in waagerechter und senkrechter Richtung (nicht diagonal), wenn das Zielfeld leer ist und auf dem übersprungenen Feld eine Kugel liegt. Die übersprungene Kugel wird dann vom Brett genommen. Gesucht wird eine Lösung, bei der die übrig bleibende Kugel am Ende in der Mitte des Spielbretts liegt.

Erstellen Sie ein Programm, das eine Zugfolge berechnet, die das Problem löst.

Modellieren Sie dazu das Spielbrett als 7x7-Array, von dem nur das "innere Kreuz" für das Spiel genutzt wird. Mein Programm findet in diesem Modell die folgende Zugfolge:

```
31: (5,3)->(3,3)
30: (4,5)->(4,3)
29: (6,4)->(4,4)
28: (6,2)->(6,4)
27: (3,4)->(5,4)
26: (6,4)->(4,4)
25: (4,3)->(4,5)
24: (4,6)->(4,4)
23: (1,4)->(3,4)
22: (4,4)->(2,4)
21: (3,2)->(3,4)
20: (1,2)->(3,2)
19: (4,2)->(2,2)
18: (4,0)->(4,2)
17: (5,2)->(3,2)
16: (3,4)->(1,4)
15: (3,6)->(3,4)
14: (0,4)->(2,4)
13: (3,4)->(1,4)
12: (3,2)->(1,2)
11: (3,0)->(3,2)
10: (0,2)->(2,2)
 9: (3,2)->(1,2)
 8: (2,6)->(2,4)
 7: (2,4)->(0,4)
 6: (2,0)->(2,2)
 5: (2,3)->(2,1)
 4: (0,4)->(0,2)
 3: (0,2)->(2,2)
 2: (2,1)->(2,3)
 1: (1,3)->(3,3)
```

Die Zugfolge wird übrigens rückwärts ausgegeben, weil ich in meinem Programm mit Rekursion arbeite und die Lösung beim Rückzug aus der Rekursion

Die Zugfolge wird übrigens rückwärts ausgegeben, weil ich in meinem Programm mit Rekursion arbeite und die Lösung beim Rückzug aus der Rekursion ausgegebe. Sie können sich ja zusätzlich überlegen, wie man eine Vorwärtsausgabe realisiert.

Feedback

Du hast diese Aufgabe am 21.11.16, 14:33 als bearbeitet markiert und dabei kein Feedback abgegeben.