

Praktikum Programmieren C/C++ Blatt 3

Aufgabe 1: Binomialkoeffizient

Der **Binomialkoeffizient** ist eine mathematische Funktion, mit der sich eine der Grundaufgaben der Kombinatorik lösen lässt. Er gibt an, auf wie viele verschiedene Arten man k Objekte aus einer Menge von n verschiedenen Objekten auswählen kann (ohne Zurücklegen, ohne Beachtung der Reihenfolge). Der Binomialkoeffizient „49 über 6“ entspricht damit beispielsweise der Anzahl der möglichen Ziehungen beim Lotto.

Ein Binomialkoeffizient hängt von zwei Zahlen n und k ab, wobei immer gelten muss: $n \geq k$. Er wird mit dem Symbol

$$\binom{n}{k}$$

geschrieben und als „ n über k “, „ k aus n “ oder „ n tief k “ gesprochen. Für nicht negative ganze Zahlen lässt sich die Funktion auf folgende Art definieren:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n - k)!}.$$

Implementieren Sie die Binomialfunktion basierend auf der rekursiven Fakultätsfunktion von Blatt 2.

Implementieren Sie eine direkt rekursive Version basierend auf folgenden Gesetzen:

$$\binom{n}{n} = \binom{n}{0} = 1; \quad \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Aufgabe 2: Fakultät iterative

Wandeln Sie die rekursive Variante der Fakultätsfunktion aus Blatt 2 in eine iterative Variante unter Verwendung einer `while`-Schleife um.

Aufgabe 3: Anagramm

Ein Anagramm bezeichnet ein Wort, das aus einem anderen Wort durch Umstellung (Permutation) der Buchstaben gebildet wurde. Zum Beispiel ist „Nana“ ein Anagramm von „Anna“. Anstatt eines einzigen Wortes kann man auch ganz allgemein die Umstellung der Zeichen einer Zeichenkette als Anagramm bezeichnen.

Eine einfache Art zu bestimmen, ob eine Zeichenkette ein Anagramm einer anderen ist, besteht darin von jeder Zeichenkette zu bestimmen, welche Zeichen sie wie oft enthält. Enthalten beide Zeichenketten dieselben Zeichen gleich oft, sind sie Anagramme. Zum Beispiel: Anna enthält jeweils „a“ und „n“ zweimal (wir ignorieren ja die Groß- und Kleinschreibung) und ebenso sind in „Nana“ „a“ und „n“ je zweimal enthalten.

Schreiben Sie eine Funktion

```
bool ist_anagramm(char *s1, char *s2)
```

die zwei überprüft, ob die beiden Zeichenketten `s1` und `s2` Anagramme sind. Dabei sollen Sie die Groß- und Kleinschreibung außer Acht lassen, d.h. ein „A“ und ein „a“ gelten als gleichwertig.

Schreiben Sie ein Hauptprogramm, das zwei Zeichenketten von der Tastatur einliest und dann ausgibt, ob diese Zeichenketten Anagramme voneinander sind.

Hinweis: Sie können für die Lösung der Aufgabe einen Vektor `int count[256]` definieren, und diesen mit dem ASCII-Code von Zeichen der eingegebenen Zeichenkette indizieren und so zählen, wie oft jedes Zeichen vorkommt. In `count['a']` würden Sie z.B. abspeichern, wie oft der Buchstabe „a“ in einer Zeichenkette auftaucht. Weiterhin können Sie die Funktion `int tolower(int ch)` aus der C Standardbibliothek verwenden, um einen Buchstaben in den korrespondierenden Kleinbuchstaben umzuwandeln. Da sie zwei Zeichenketten einlesen, brauchen Sie u.U. auch zwei solche Vektoren (es geht allerdings auch mit einem, wenn Sie geschickt vorgehen).

Aufgabe 4: Feld

Schreiben Sie eine Funktion

```
bool is_in(int x, int f[MAX], int *index)
```

die überprüft, ob die Zahl `x` in dem Feld `f` enthalten ist. Über den Referenzparameter `index` soll die Position des ersten Vorkommens zurückgegeben werden.

Testen Sie die Funktion in einem Hauptprogramm.