

Programmierung Numerischer Algorithmen in C++

Übungsblatt 3

SS 2019

Übungsaufgabe 11

Das Pascalsche Dreieck ist eine geometrische Darstellung der Binomialkoeffizienten $\binom{n}{k}$. Sie sind im Dreieck so angeordnet, dass jeder Eintrag die Summe der zwei darüberstehenden Einträge ist:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
  ⋮

```

- Man schreibe eine Funktion `void pascal(int* P, int n)`, die die n -te Zeile des Pascalschen Dreiecks berechnet. Der Funktion werden ein Zeiger auf ein eindimensionales Feld `P` (der Länge $> n$) und die Zeile n , die berechnet werden soll übergeben. In der Funktion selbst soll kein weiteres Feld erzeugt werden, sondern das Feld `P` immer wieder überschrieben werden. Man achte darauf, dass das Feld von "hinten" überschrieben wird, da sonst die zur Berechnung notwendigen Daten schon überschrieben sind.
- Man erweitere die Funktion `pascal` zu der Funktion `void fibo(int* P, int* F, int n)`, die aus `P` die ersten n Fibonacci-Zahlen berechnet und in dem Feld `F` speichert. Die Fibonacci-Zahlen können aus dem Pascalschen-Dreieck berechnet werden, indem die Zahlen auf den Diagonalen aufsummiert werden:

```

      1
     ↗ 1 2
    ↗ ↗ 3 5
   ↗ ↗ ↗ 8
  ↗ ↗ ↗ ↗ ⋮
 1 2 1 ↗
 1 3 3 1
 1 4 6 4 1
 1 5 10 10 5 1
    ⋮

```

Übungsaufgabe 12

- a) Man schreibe eine Funktion `char* find(char c, char* p)`, der ein Zeichen `c` und ein nullterminiertes `char`-Feld `p` (als Zeiger) übergeben wird, und die einen Zeiger auf das erste `c` in diesem Feld zurückgibt.
- b) Schreiben Sie eine Funktion `void invert(char* p)`, der ein `char`-Feld `p` als Zeiger übergeben wird. Die Funktion soll das `char`-Feld so verändern, dass die Einträge in umgekehrter Reihenfolge in diesem Feld stehen, also z.B. aus "abcde" wird "edcba". Hierbei verwende man die Array-Notation `p[index]`.
- c) Man verändere die Funktion `invert` so, dass ausschließlich die Zeiger-Notation verwendet wird.

Übungsaufgabe 13

Man schreibe ein Programm, das den Operator `+` neu definiert, sodass wenn wir zwei `int` Zahlen `a` und `b` summieren, erhalten wir `a*a-b*b` und wenn wir zwei `double` Zahlen `d` und `e` summieren, erhalten wir `d*d+e*e`. Testen Sie das Programm.

Übungsaufgabe 14

Man beseitige alle Fehler und Warnungen in dem Programm `Prog_C++2019_14-wrongsyntax.cpp`. Was ist der Grund für diese Meldungen? Verbessere man den Code, damit alle Befehle in `main` ausgeführt werden können.

Übungsaufgabe 15

Schreiben Sie drei Funktionen

```
double horner1(int n, double a[], double x),  
void horner2(int n, double a[], double x, double* res) und  
void horner3(int n, double a[], double x, double& res),
```

die das Horner-Schema $p(x) = ((a_n x + a_{n-1})x + \dots + a_1)x + a_0$ für die Auswertung eines Polynoms $p(x)$ implementieren, vgl. Sie mit der Vorlesung, Seite 45.

Testen Sie diese Funktionen mit statischer und dynamischer Speicherallokation für `a[]`.

Erklären Sie die Ergebnisse und die Syntax.