



Übungen zur Vorlesung Modellierung und Programmierung WS 2015–2016

Blatt 6

Aufgabe 1 (1+2+2=5 Punkte)

Implementieren Sie eine Struktur für symmetrische Matrizen. Gehen Sie hinsichtlich der Indizierung wie in der Vorlesung gezeigt vor. Schreiben und testen Sie die folgenden Funktionen:

- Erzeugen und Löschen einer symmetrischen Matrix, Ein- und Ausgabe aus bzw. in Dateien (die zeichenorientierte Form genügt).
- Summe zweier symmetrischer Matrizen, Produkt aus einer symmetrischen Matrix und einem Vektor.
- Produkt zweier symmetrischer Matrizen. Überlegen Sie sich anhand eines einfachen Beispiels, dass das Ergebnis im allgemeinen nicht symmetrisch ist. Verwenden Sie zum Speichern des Resultats daher die allgemeine Matrix-Struktur.

Testen Sie ihre Funktionen wie immer in einem Hauptprogramm.

Aufgabe 2 (3+(5)+2=5(10) Punkte)

Schreiben Sie ein Programm, das die Ziehung der Lottozahlen "6 aus 49" nachbildet: Es sollen 6 natürliche Zahlen zufällig (z.B. mit `random()`) ohne Zurücklegen gezogen werden. Wie üblich sollen die Lottozahlen aufsteigend sortiert ausgegeben werden.

- Implementierung mit Feldern.* Die Gewinnzahlen werden in einem Feld gespeichert. Um festzustellen, ob eine Zahl bereits gezogen wurde, genügt die Verwendung der einfachen Suche aus der Vorlesung. Wie üblich sollen die Lottozahlen aufsteigend sortiert ausgegeben werden. Verwenden Sie zum Sortieren den einfachen Algorithmus aus der Vorlesung.
- Implementierung mit Listen (optional).* Die Lottozahlen und die Gewinnzahlen werden in zwei Listen gespeichert. Man zieht ein Zufallselement aus der Liste mit Lottozahlen und fügt sie in die Liste mit Gewinnzahlen ein. Die Einfügung muss so implementiert werden, dass die Liste immer sortiert bleibt.
- Erweitern Sie die Implementierungen (a) und (b) so, dass sie auch für die Ziehung der m aus n Lottozahlen funktionieren. Schätzen Sie die Speicher- und Laufzeit-komplexität beider Implementierungen ab.

Aufgabe 3 (5 Punkte)

Implementieren Sie den in der Vorlesung gezeigten stabilisierten Algorithmus zur Lösung quadratischer Gleichungen als Funktion. Testen Sie die Funktion, indem Sie sie in das Programm aus Aufgabe 1, Übungsblatt 2 integrieren und die jeweils berechneten Nullstellenwerte miteinander vergleichen.

Aufgabe 4 (5 Punkte)

Betrachten Sie die Struktur

```
struct student {
    char *Name;
    char *Vname;
    unsigned MatrNr;
};
```

aus der Vorlesung, wobei für diese Aufgabe nur ein Vorname zulässig sein soll. Legen Sie mit Hilfe einer Eingabedatei ein unsortiertes Feld an, dessen Einträge vom obigen Datentyp sind. Sortieren Sie dieses Feld

- aufsteigend nach Matrikelnummer,
- alphabetisch aufsteigend nach (Nach-)Namen.

Die Sortierung soll mit Hilfe des Quicksort-Algorithmus durchgeführt werden. Implementieren Sie diesen jedoch nicht selbst, sondern verwenden Sie die Bibliotheksfunktion `qsort()`, indem Sie Zeiger auf die jeweils zum Einsatz kommende Vergleichsfunktion einsetzen.

Im Hauptprogramm soll ferner ein Name von der Tastatur eingelesen und im Anschluss mit Hilfe der Bibliotheksfunktion `bsearch()` festgestellt werden, ob sich ein Eintrag dieses Namens im sortierten Feld befindet.

Abgabe der Lösungsvorschläge und Vorführung der praktischen Aufgabe vor dem 10.02.2016.