

**Imperative Programmierung**  
**Aufgabenblatt A04**  
**Ausgabe: 9. Dezemer 2019**  
**Abgabe bis: 22. Dezember 2019, 23:59**

---

**Hinweise:** Die Lösungen der Aufgaben sind als *PDF-Dokument* bzw. *C-Quelltext* mit Hilfe des Versionsverwaltungssystems Subversion (SVN) abzugeben. Platzieren Sie ein PDF-Dokument mit Ihren Antworten im Ordner A04 innerhalb Ihres Gruppenverzeichnisses<sup>1</sup>. Platzieren Sie die C-Quelltexte im Unterordner A04/src. Schreiben Sie in die abgegebenen Dateien Ihren Namen und Ihre Matrikelnummer. Die Referenzsysteme für die Kompilierung sowie Ausführung der Programme sind der PC-Pool und der ITMZ Remote Server (UniComp). Ihre Quelltexte müssen mit den Compilerflags „-ansi -Wall -pedantic“ ohne Warnungen und Fehler übersetzbar sein.

**Verspätete Abgaben oder Abgaben ohne Matrikelnummer werden nicht gewertet!**

---

1. Schreiben Sie ein rekursives Programm, das alle Permutationen von  $n$  Elementen ausgibt. (D.h., alle möglichen unterschiedlichen Reihenfolgen der Elemente.) Das Programm soll die Zahl  $n$  als Kommandozeilenargument akzeptieren (Sie können die Methode hierfür von `hanoi.c` kopieren). Sie können davon ausgehen, dass  $n$  nicht größer als 26 ist, so dass Sie die Elemente mit Buchstaben bezeichnen können. Ihr Programm soll damit beispielsweise folgende Ausgabe liefern:

```
$ ./perm 3
1: ABC
2: BAC
3: CAB
4: ACB
5: BCA
6: CBA
6 permutations
$
```

Tipp: Heap's Algorithmus.

12 Punkte

2. Sortieren. Entwickeln Sie eine Funktion zum Sortieren eines Feldes von Zahlen mit Hilfe der Funktion `swap` aus der Vorlesung.

Die Idee ist einfach:

- Wenn zwei benachbarte Zahlen im Feld in der falschen Reihenfolge sind, dann werden diese vertauscht.
- Dies machen Sie so lange, bis keine Vertauschungen mehr erforderlich sind.

15 Punkte

---

<sup>1</sup>Ihr Gruppenverzeichnis finden Sie unter  
<https://svn.informatik.uni-rostock.de/lehre/ip2019/groups/IhrGruppenkürzel>.

3. Betrachten Sie folgendes Code-Fragment:

```
#define N 29
#define C_SPACE 26
#define C_COMMA 27
#define C_STOP 28

int getcc() {
    int c, haveSpace = 0;

    while(isspace(c=getchar())) haveSpace = 1;

    if(haveSpace) return (ungetc(c,stdin),C_SPACE);
    else if(c>='a' && c<='z') return c-'a';
    else if(c>='A' && c<='Z') return c-'A';
    else if(c==',' ) return C_COMMA;
    else if(c=='.' ) return C_STOP;
    else if(c==EOF) return EOF;
    else return getcc();
}
```

Erläutern Sie, was dieser Code macht. Falls Ihnen die Funktionen `isspace` und `ungetc` nicht bekannt sind, recherchieren Sie, was diese Funktionen tun.

5 Punkte

4. Schreiben Sie ein Programm, das Text in „Pseudo-Englisch“ erzeugen kann. Dazu gehen Sie folgendermaßen vor:

- Bauen Sie ein dreidimensionales Feld auf, das für jedes Zeichen  $c$  (Buchstaben sowie Leerzeichen, Komma, und Punkt) die bedingte Wahrscheinlichkeit beinhaltet, dass das Zeichen  $c$  auf die Zeichen  $a$  und  $b$  folgt.

Diese Tabelle können Sie beispielsweise durch die Analyse von Herman Melvilles „Moby Dick“ erzeugen, diesen finden sie auf der Stud.IP-Seite der Vorlesung.

10 Punkte

- Entwickeln Sie ein Verfahren, mit dem Sie aus einem Vektor von Werten  $(v_1, \dots, v_n)$ , für die Sie einen Vektor von Wahrscheinlichkeiten  $(p_1, \dots, p_n)$  gegeben haben, einen Wert  $v_i$  zufällig wählen können, so dass die Wahrscheinlichkeit, dass Sie  $v_i$  bekommen eben genau  $p_i$  entspricht. Sie können davon ausgehen, dass gilt

$$\sum_{i=1}^n p_i = 1$$

10 Punkte

- Erzeugen Sie mit Hilfe des dreidimensionalen Feldes einen Zufallstext dadurch, dass Sie sich jeweils die beiden zuletzt generierten Zeichen  $a$  und  $b$  merken und dann ein neues Zeichen  $c$  zufällig wählen, wobei Sie die Wahrscheinlichkeit eines bestimmten Zeichens  $c$  gegeben die Vorgänger  $a$  und  $b$  aus dem Feld entnehmen. Um das erste Zeichen zu erzeugen können Sie annehmen dass die Vorgängerzeichen der Punkt und das Leerzeichen sind.

10 Punkte