

## 4. Übungsblatt zur Vorlesung „Einführung in die Programmiersprache C++“

### Aufgabe 4.1: Pointerarithmetik (35 Punkte)

**Neue Konzepte dieser Aufgabe:** Operationen mit Pointer-Variablen und Arrays

Es seien `feld` ein `int`-Array `p`, `p1`, `p2` `int`-Zeiger und `i` eine `int`-Variable. Welche der folgenden Zuweisungen sind zulässig und welche nicht? Begründen Sie ihre Antwort!

- |                                   |                                 |                              |
|-----------------------------------|---------------------------------|------------------------------|
| a) <code>p = feld;</code>         | d) <code>feld[2] = p[5];</code> | g) <code>i = p1 * p2;</code> |
| b) <code>feld = p;</code>         | e) <code>p1 = p2 + i;</code>    | h) <code>i = p1 - p2;</code> |
| c) <code>p = &amp;feld[3];</code> | f) <code>p1 = i + p2;</code>    | i) <code>i = p1 + p2;</code> |

Erklären sie darüber hinaus, wie der `[][]`-Zugriff bei zweidimensionalen Arrays ohne die Verwendung von `[]` – d.h. über die Verwendung der Pointer direkt – realisiert werden kann. Eine nette Möglichkeit verwirrenden Programmcode zu generieren ist es, Array-Namen und Index zu vertauschen, d.h. `feld[i]` stellt denselben Zugriff wie `i[feld]` dar. Warum ist das so? Geben sie Ihre Antworten und Begründungen schriftlich ab.

### Aufgabe 4.2: Speicherdebugging mit `valgrind` (35 Punkte)

Das Tool `valgrind` ist ein Programm zur Überprüfung von Speicherzugriffen in C/C++-Programmen. Es trackt die Allokierung von dynamischem Speicher und kann somit verschiedene Fehlerquellen im Speichermanagement aufspüren. Der Nachteil ist, dass die Programme dadurch sehr viel langsamer (Größenordnung Faktor 10) werden. Machen Sie sich mit der Verwendung von `valgrind` vertraut. Was bedeuten die Angaben „definitely lost“, „indirectly lost“, „possibly lost“, still reachable“ und „suppressed“. Geben Sie jeweils ein Beispiel dafür an, unter welchen Umständen diese Fehler auftreten können.

In Ihrem Repository finden Sie im Ordner `valgrind/` verschiedene Beispielprogramme. Übersetzen Sie diese und führen Sie sie mit `valgrind` aus. Erklären sie mögliche Fehlerquellen und die entsprechenden Fehlermeldungen von `valgrind`.

### Aufgabe 4.3: Debugging mit `gdb` (30 Punkte)

Machen Sie sich mit dem GNU-Debugger `gdb` vertraut. Erklären Sie, wie sie im Kommandozeileninterface des `gdb`

- Einen Backtrace durchführen
- Werte von Variablen in einem Kontext ausgeben können
- Breakpoints und Conditional-Breakpoints setzen
- Durch das Programm steppen können (Zeilenweise und Blockweise)

In Ihrem Repository finden Sie im Ordner eine fehlerhafte Implementierung einer verketteten Liste in C++. Finden Sie den Fehler mit `gdb` und erklären Sie Ihrem Tutor, wie sie ihn gefunden haben. Schreiben Sie ein `cmake`-Skript zum Übersetzen des Programms mit passenden `g++`-Flags für Debug- und Release-Mode. Erklären Sie, wie Sie zwischen den Modi wechseln können und warum Sie welche Parameter gesetzt haben.

**Abgabe:**

Checken Sie ihre Abgabe bis Montag, 21.11.2022, 08:00 Uhr in den Ordner `uebung04` des Master-Branches Ihres git-Repositories ein.