

**Universität Potsdam
Institut für Informatik**

Praxis der Programmierung

5. Aufgabenblatt

1. Kopieren Sie die Datei `typfehler.c` zur Behandlung von Typfehlern durch ungeeignete Benutzereingaben mittels `scanf`. Ändern Sie das Programm so, dass eine Fehlerbehandlung mit Recovering erfolgt.
2. Schreiben Sie ein C-Programm `pkub.c`, das mit einem Parameter gestartet werden muss (sonst erfolgt eine Fehlermeldung) und dann die Kubikzahl des Parameters als `double`-Zahl ermittelt und auf `stdout` ausgibt.
3. Definition einer Struktur von Punkten in der Ebene
 - a) Kopieren Sie die Datei `point_1.c` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler. Analysieren Sie den Quellcode.
 - b) Kopieren Sie `point_1.c` in `point_2.c` und editieren Sie die Kopie. Die Datei `point_2.c` soll eine Funktion zum Verschieben auf neue Koordinaten enthalten. Testen Sie diese Funktion. Überprüfen Sie dabei, ob beim Initialisieren einer Strukturvariable mit einer anderen (bereits existierenden) Strukturvariablen alle Member-Werte im Speicher kopiert werden oder ob nur ein Pointer übergeben wird. *Wie kann man das überprüfen?*
 - c) Kopieren Sie die Datei `point_3.c` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler. Analysieren und erklären Sie den Quellcode.
4. Ein *Polygonenzug* ist eine zusammenhängende Folge von Strecken, der durch eine Liste von Punkten spezifiziert werden kann.
 - a) Kopieren Sie die Datei `poly.c` aus Moodle oder von `/home/rlehre/pdp/2025/u05`. Vervollständigen Sie diese um die Implementierung eines Polygonenzugs als verkettete Liste von Punkten in der Ebene (aus Aufgabe 3). Richten Sie sich dabei nach der Definition von Listen und Listenelementen aus der Vorlesung und beachten Sie, dass Ihre Definition zur gegebenen Definition der `append`-Funktion passt. Diese dient dazu am Ende eines Polygonenzugs ein neues Element anzufügen.
 - b) Definieren Sie die Funktion `pretty_print` zum Ausgeben eines Polygonenzugs durch Ausgabe seiner Punkte auf die Konsole.
 - c) Testen Sie die Funktionen `append` und `pretty_print`. Finden und beheben Sie den logischen Fehler in `append`. Nutzen Sie den Debugger `gdb` um herauszufinden, wo der `Segmentation Fault` ausgelöst wird.

5. Vervollständigen Sie das Programm um die folgenden Operationen:

- **shorten** zum Verkürzen eines Polygonenzugs durch Entfernen des letzten Punkts aus der Liste,
- **insert** zum Einfügen eines neuen Punkts an einer bestimmten Position (Index) in der Liste,
- **mirror**, die einen neuen Polygonenzug erzeugt, in dem alle Punkte des übergebenen Polygonenzugs in umgekehrter Reihenfolge auftreten.

Testen Sie Ihre Implementierung, in dem Sie nacheinander

- einen leeren Polygonenzug anlegen,
- zwei Punkte hinten anfügen,
- je einen Punkt an der ersten und dritten Stelle einfügen,
- einen neuen Polygonenzug erzeugen, in dem alle Elemente des ersten Polygonenzugs in umgekehrter Reihenfolge auftreten, und
- alle Elemente aus dem ersten Polygonenzug nacheinander entfernen.

Lassen Sie vor und nach jeder Operation den Polygonenzug ausgeben.