

Universität Potsdam
Institut für Informatik
Praxis der Programmierung

Aufgabenblatt 1b

1. Kopieren Sie die Datei `typengroesse.c` aus Moodle.
- a) Analysieren Sie die Syntax und Bedeutung des `sizeof`-Operators. Welche Operanden können auf `sizeof` folgen? Was wird zurückgegeben?

- b) Ergänzen Sie das Programm, so dass Sie eine Übersicht über die Größe der Speicherbereiche für jeden der elementaren Datentypen erhalten.

2. Kopieren Sie die Datei `kreisberechnung.c` aus Moodle oder von `rlehre` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler. Das Programm soll den Umfang und die Fläche eines Kreises für einen eingegebenen Radius `r` berechnen.

- a) Warum werden die Werte für Fläche und Umfang nicht richtig ausgegeben? Beschreiben Sie den Fehler.

Nennen Sie einen Ansatz, um den Fehler zu beheben, und setzen Sie ihn um.

- b) Führen Sie das berichtigte Programm aus und geben Sie als Testwert 50000 ein. Welche Fehler treten nun auf? Wie kommen diese zu Stande?

Nennen Sie einen Ansatz, um den Fehler zu beheben, und setzen Sie ihn um.

3. Kopieren Sie die Datei `insertionsort.c` aus Moodle oder von `rlehre` in Ihr Arbeitsverzeichnis und übersetzen Sie sie mit einem C-Compiler.

Im Programm wurde eine Funktion umgesetzt, die Arrays mit Hilfe von Insertionsort sortieren soll. Nutzen Sie den `gdb`-Debugger, um herauszufinden, warum die Sortierung nicht funktioniert. Erklärungen zu Insertionsort und `gdb` finden Sie weiter unten.

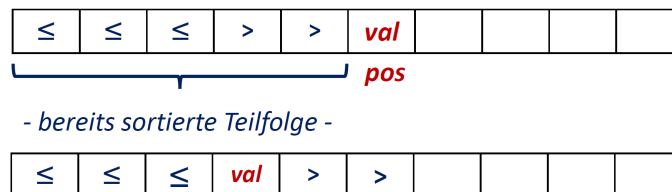
- a) Kompilieren Sie das Programm mit `gcc -g insertionsort.c -o insertionsort`, um es fürs Debugging vorzubereiten.
- b) Starten Sie mit `gdb insertionsort` den Debugger. (Verlassen mit `quit`)
- c) Legen Sie fest, in welchen Codezeilen Breakpoints platziert werden müssen. Begründen Sie Ihre Angaben kurz.

- d) Platzieren Sie die Breakpoints mit `break codezeile`.
- e) Starten Sie das Programm mit `run` und finden Sie heraus, wie es abläuft. Lassen Sie sich an den Breakpoints die Werte der Variablen ausgeben, um den Zustand des Programms zu verfolgen.
- f) Finden und beheben Sie den Fehler.

Insertionsort

Insertionsort ist ein Sortieralgorithmus, bei dem, wie der Name schon vermuten lässt, die Werte am richtigen Platz eingefügt werden. Dabei wird davon ausgegangen, dass der vordere Teil der zu sortierenden Liste (bis vor die Position *pos*) bereits sortiert ist. Zu Beginn besteht dieser sortierte Teil nur aus einem Element und eine einelementige Liste ist natürlich immer sortiert.

Beim Sortieren wird der Wert *val* des ersten Elements hinter dem sortierten Bereich (also des Elements an Position *pos*) in den sortierten Bereich einsortiert. Die größeren Elemente im sortierten Bereich rutschen dabei um eine Stelle weiter nach rechts, sodass der sortierte Bereich um den Platz des betrachteten Elements wächst. Dies wird mit dem nächsten Element außerhalb des sortierten Bereichs wiederholt, bis die gesamte Liste sortiert ist.



Debuggen mit gdb

Der GNU Project Debugger (für C, C++ und Python)

Kompilieren, Starten, Verlassen

Kompilieren Sie das Programm mit `gcc -g insertionsort.c -o insertionsort`. Der Parameter `-g` bereitet das Debugging vor, indem entsprechende Informationen produziert werden, mit denen GDB arbeitet.

Starten Sie mit `gdb insertionsort` den Debugger für das ausführbare Programm `insertionsort`. Sie können den Debugging-Modus mit `quit` wieder verlassen.

Befehle

<code>run</code>	Führe das Programm aus. Stoppe beim ersten Breakpoint.
<code>start</code>	Führe das Programm aus. Stoppe vor der <code>main</code> -Methode.
<code>break <linenumber></code>	Platziere einen Breakpoint in Codezeile <code>linenumber</code> .
<code>break <function></code>	Platziere einen Breakpoint am Anfang der Funktion <code>function</code> .
<code>print <varname></code>	Zeige den aktuellen Wert der Variable <code>varname</code> an.
<code>watch <varname></code>	Unterbrich, wenn der Wert von <code>varname</code> verändert wird, und zeige alten und neuen Wert an.
<code>continue</code>	Führe das Programm bis zum nächsten Breakpoint fort.
<code>clear <linenumber></code>	Lösche den Breakpoint in Codezeile <code>linenumber</code> .
<code>clear <function></code>	Lösche den Breakpoint der Funktion <code>function</code> .
<code>delete</code>	Lösche alle Breakpoints.
<code>help</code>	Öffne die Hilfe-Page.