

Algorithmen und Programmierung

Einfache Ein- und Ausgabe (Trainingsaufgabe)

In der *Hello World*-Trainingsaufgabe haben Sie bereits gesehen, dass es möglich ist, Ausgaben in einem Terminal zu erzeugen. Dies ist mittels der Funktion `printf` geschehen. Analog dazu existiert die Funktion `scanf`, mit welcher Eingaben während des Programmflusses getätigt werden können. Beide unterstützen Formatierungsanweisungen. Diese beginnen stets mit % stellen Platzhalter für ein-/auszugebende Daten dar. So kann beispielsweise für ganze Zahlen `%d` verwendet werden. Für einen Überblick über weitere Möglichkeiten sei hier z.B. auf `man 3 printf` verwiesen. Jedoch verwendet `scanf` den sogenannten Referenzierungsoperator, daher sei für ganze Zahlen die folgende Hilfsfunktion gegeben:

```
1  int get_number()
2  {
3      int res;
4      scanf("%d", &res);
5      return res;
6  }
```

Dies ist aber nicht der einzige Weg, Eingaben an ein Programm zu übergeben. Weiterhin können die Parameter der `main`-Funktion dazu genutzt werden:

```
1  int main(int argc, char* argv[]);
```

Dabei repräsentiert `int argc` die Anzahl der übergebenen Programmparameter, welche sich hinter `char* argv[]` verbergen. Anzumerken ist dabei, dass hinter `argv[0]` sich der Programmname des Aufrufs verbirgt.

Zur Konvertierung eines solchen Programmparameters (oder allgemein einer Zeichenkette) kann für ganze Zahlen die Funktion `int atoi(char *str)` verwendet werden. Diese liefert den numerischen Wert der Zeichenkette zurück: `atoi("42") == 42`.

Aufgabe:

Schreiben Sie eine Quelltextdatei `compare.c`, deren Aufgabe es ist, Eingaben des Nutzer gegen den ersten übergebenen Programmparameter als Ganzzahlen zu vergleichen. Dabei soll nur zwischen „kleiner als“ ($<$), „gleich“ ($=$) und „größer als“ ($>$) unterschieden werden.

Beispiele:

1.
 - Programmaufruf `./compare 42`
 - Eingabe: 23
 - Ausgabe: `42 > 23`
2.
 - Programmaufruf `./compare 0`
 - Eingabe: 0
 - Ausgabe: `0 = 0`
3.
 - Programmaufruf `./compare -5`
 - Eingabe: 5
 - Ausgabe: `-5 < 5`

Achten Sie darauf, dass die Datei zur Abgabe eine `main`-Funktion bereitstellen muss.

Hinweise zur Aufgabenstellung

Für die Lösung dieser Aufgabe benötigen Sie folgende Grundkenntnisse:

- Kontrollfluss in C (`if`)
- Verwendung von Kommandozeilenparameter
- Lesen und Verarbeiten von Eingaben
- Schreiben einer `main`-Funktion
- Benutzung von `gcc`

Hinweise zur Abgabe

- Erstellen Sie eine ZIP- bzw. TGZ-Archivdatei, welche die geforderten Dateien enthält.
- Fügen Sie dem Archiv keine weiteren Dateien oder Ordner hinzu.
- Reichen Sie Ihre Lösung unter <https://osg.informatik.tu-chemnitz.de/submit> ein.
- Bis zum Abgabende (Deadline), sofern gegeben, können beliebig neue Lösungen eingereicht werden, die die jeweils älteren Versionen ersetzen.
- Ihr Programm muss auf der Testmaschine übersetzbar sein. Deren Details sind auf dem OpenSubmit-Dashboard verfügbar.
- Ihre Lösung wird automatisch validiert. Sie werden über den Abschluss der Validierung per eMail informiert.