Einführung in die Programmierung, WS 2016/2017
Blatt 5

Simon Hangl, Sebastian Stabinger, Alex Hirsch

2016–11–22

- Abgabe bis spätestens Montag 21:59 über OLAT (`https://lms.uibk.ac.at/olat/dmz/`).
- Bereiten Sie jede Aufgabe so vor, dass Sie Ihre Lösung im Proseminar präsentieren können!
- Benennen Sie Ihre Abgabe nach folgendem Schema: *Gruppennummer*-*Nachname*-`blatt`*Übungsblattnummer*`.tar.gz` Wenn Sie also Max Mustermann heißen und Gruppe 1 besuchen, heißt die Datei von Übung 5: `1-mustermann-blatt5.tar.gz`
- Compilieren Sie alle Programme mit den Optionen `-Wall -Werror -std=c99`

Feedback

Nutzen Sie die angebotenen Möglichkeiten, uns Feedback zu geben (eMail, Tutorium, Proseminar). Hier können Sie uns auf Probleme, notwendige Stoffwiederholungen, Unklarheiten, aber auch positive Dinge, die beibehalten werden sollten, hinweisen.

Testen und Dokumentation

Stellen Sie sicher, dass alle Lösungen fehlerfrei kompilieren. Testen Sie Ihre Lösungen ausführlich (z.B. auf falsche Eingaben, falsche Berechnungen, Sonderfälle) und dokumentieren Sie sie. Dies hilft Ihnen bei der Präsentation und uns beim Nachvollziehen Ihrer Entscheidungen. Lesen Sie die Aufgaben *vollständig* durch.

Aufgabe 1 (4 Punkte)

Implementieren Sie folgendes Spiel: Der Computer *überlegt* sich eine Zufallszahl zwischen 1 und 100.

Anschließend werden vom Benutzer drei Zahlen abgefragt. Ist die gesuchte Zahl (die vorher generierte Zufallszahl) dabei, gewinnt der Spieler und das Programm endet. Ist die richtige Zahl nicht dabei, erfährt der Spieler ob die Mehrheit der Zahlen größer oder kleiner als die gesuchte Zahl ist oder ob ein Unentschieden besteht. Zusätzlich soll noch der kleinste der Abstände zwischen den 3 Zahlen und der gesuchten Zahl ausgegeben werden. Im Anschluss werden erneut 3 Zahlen vom Benutzer abgefragt. Dies wiederholt sich solange der Spieler die gesuchte Zahl nicht erraten hat.

Hinweis: Sie können mit `srand(time(NULL))` einen Zufallszahlengenerator initilisieren und anschließend mit `(rand() % 100) + 1` eine Zufallszahl zwischen 1 und 100 erzeugen. Dazu müssen die Header `stdlib.h` und `time.h` eingebunden werden.

Hinweis: *Abgabe: 1-mustermann-a1.c*

Aufgabe 2 (6 Punkte)

Es sollen Funktionen zum Lösen von quadratischen Gleichungen implementiert werden. Eine quadratische Gleichung ist gegeben durch

$$ax^2 + bx + c = 0 \tag{1}$$

In der `main`-Funktion sollen die 3 Zahlen a, b, c von der Konsole eingelesen werden. Schreiben Sie eine Funktion

```
double compute_discriminant(double a, double b, double c)
```

die die Diskriminante $D = b^2 - 4ac$ berechnet und zurückgibt. Schreiben Sie eine weitere Funktion `is_solvable`, die zurückgibt, ob die quadratische Gleichung reelle Lösungen hat ($D >= 0$). Überlegen Sie sich welche Parameter und Rückgabewert die Funktion hat.

Schreiben Sie weiters zwei Funktionen

```
double compute_x1(double a, double b, double c)
```

```
double compute_x2(double a, double b, double c)
```

die die Lösungen x_1, x_2 der quadratischen Gleichung berechnet. Sollte keine reelle Lösung existieren, sollen die Funktionen 0 zurückgeben. Dazu sollen die Funktionen die bereits geschriebenen Funktionen `compute_discriminant` und `is_solvable` verwenden.

Schreiben Sie ein **ausführliches** Testprogramm, das die Grenzfälle berücksichtigt. Welche Grenzfälle kann es geben? Nennen Sie mindestens 3.

Hinweis: *Abgabe: 1-mustermann-a2.c*

Aufgabe 3 (4 Punkte)

Implementieren Sie einen kommandozeilenbasierten Taschenrechner. Fragen Sie zuerst eine Operation von der Kommandozeile ab (addieren, subtrahieren, multiplizieren, beenden). Wählt der Benutzer beenden, bricht das Programm ab. Andernfalls werden zwei Operanden von der Kommandozeile abgefragt. Geben Sie das Ergebnis der Operation aus und wiederholen Sie den Vorgang (Operation abfragen, Operanden einlesen, etc.).

Implementieren Sie zumindest ein minimales Kommandozeilen-Interface, das dem Benutzer anzeigt welche Eingabe gerade erwartet wird, und welche Optionen verfügbar sind.

Hinweis: *Benutzereingaben können mit scanf abgefragt werden.*
Hinweis: *Abgabe: 1-mustermann-a3.c*