

Übung 1

Abgabe: 17.04.2023, 8:00 Uhr

Hinweise zu den Übungen

Herzlich willkommen bei den Übungen zur Vorlesung „C-Programmierung für Algorithmen und Datenstrukturen“. In den Übungen wenden Sie die Vorlesungsinhalte an und erarbeiten die Klausurzulassung. Die Klausurzulassung besteht aus zwei Teilen:

- **Abgabe der Übungsblätter:** Um die Klausurzulassung zu erhalten müssen Sie insgesamt (d.h. in der Summe) mindestens 50% der möglichen Übungspunkte erreichen. Es gibt 10 Übungsblätter mit jeweils 30 Punkten, so dass Sie 150 Punkte für die Klausurzulassung benötigen.
- **Bearbeitung der Quizzes:** Zu jeder Vorlesung gibt es ein kleines Quiz, das jeweils bis zur nächsten Vorlesung (also in der Regel eine Woche) verfügbar ist. Sie müssen *in der Summe über alle Quizzes* mindestens 50% der Punkte der Quizzes erreichen.

Hinweise zur Abgabe der Übungsaufgaben Bei den Übungsaufgaben handelt es sich um Programmieraufgaben, die Sie eigenständig lösen sollen. Ihre Lösungen werden von einem automatischen System vorkorrigiert: Dazu laden Sie Ihre Lösungen bei github.com in einem Git-Repository hoch. Auf diese Weise erhalten Sie direkt bei der Abgabe Feedback über die zu erwartende Punktzahl. Bis zur Deadline dürfen Sie beliebig oft nachbessern – es zählt die Version, die bei der Deadline im Repository liegt!

Wichtig: Ihre Abgaben werden anhand von Testfällen bewertet; je mehr Testfälle Sie erfüllen, je mehr Punkte bekommen Sie. Dabei gilt: **Programme, die nicht kompilieren, werden mit 0 Punkten bewertet.** Ausnahmen von dieser Regel sind explizit gekennzeichnet.

Datenschutz: Für die Bearbeitung der Übungsaufgaben benötigen Sie einen Github-Account. Da es sich bei Github um eine in den USA ansässige Firma handelt, ist ggf. Vorsicht mit persönlichen Daten geboten. Einige Tipps dazu:

- Sie können für den Githubaccount eine kostenlose Wegwerfadresse (z.B. von <https://proton.me>) anstelle Ihrer regulären Emailadresse verwenden.

- Wenn Sie von zu Hause aus arbeiten, können Sie Ihre IP über das Uni-VPN verschleiern.
- Ein privates Browserprofil und geeignete Browsererweiterungen erschweren das Tracking.

Für Details finden Sie im ILIAS-Kurs eine Datenschutzerklärung. Sollten Sie dennoch Bedenken haben, melden Sie sich gerne.

Wie gebe ich meine Bearbeitung des Übungsblattes ab?

Sie finden in der [Mediathek eine Videoanleitung zur Abgabe](#). Vor der ersten Abgabe:

1. Vor der ersten Abgabe müssen Sie einen Github-Account erstellen. Für folgende Abgaben verwenden Sie bitte den gleichen Github-Account.
2. Wenn Sie Ihren Github-Account erstellt haben [nehmen Sie an der Umfrage im ILIAS-Kurs teil](#) und teilen uns damit ihren Githubaccount mit. Die Zuordnung des Github-Accounts zu Ihrer ILIAS-Kennung wird nur von uns intern verwendet und geht nicht an Github.

Um nun eine Abgabe für eine Aufgabe hochzuladen:

1. Klicken Sie im PDF des Übungszettels auf den Abgabelink zu der Aufgabe, die Sie bearbeiten möchten.
2. Wählen Sie „Accept assignment“ und warten Sie kurz. Github erzeugt nun ein persönliches Git-Repository für Ihre Abgabe.
3. Sobald das Repository erzeugt wurde, können Sie es auf Ihren lokalen Computer klonen oder direkt mit „Add file“ → „Upload Files“ auf der Webseite eine Datei hochladen.
4. Nach jedem Hochladen (bzw. jedem **push**) führt Github automatisch die Vorkorrektur aus. Sie sehen das Ergebnis bei „Voraussichtliche Punktzahl“.
5. Sie dürfen bis zur Deadline beliebig oft Dateien hochladen. Es zählt immer die Version, die bei der Deadline im Repository liegt.
6. Ändern Sie nur die von der Aufgabe vorgesehenen Dateien!
7. Nach Ablauf der Deadline kontrollieren unsere Korrekteure die automatische Bewertung und passen ggf. Ihre Punktzahl an. Sie erhalten die Bewertung in einem Issue in Ihrem Github-Repository.
8. Sie finden Ihre Github-Repositories unter <https://github.com/aldat-in-c> wieder. Um Ihre Repositories zu sehen, müssen Sie eingeloggt sein.

► **Aufgabe 1.1** [10 Punkte, [Link zur Abgabe](#)] Schreiben Sie ein C-Programm, das für die Zahlen $i = 1, \dots, 12$ jeweils i , i^2 und $1/i$ tabellarisch auf der Konsole ausgibt. Die Ausgabe soll *genau* folgendem Format folgen:

Zahl	Quadrat	Kehrwert
1	1	1.000000
2	4	0.500000
...
12	144	0.083333

Die Ausgabe soll also eine Tabelle mit drei Spalten enthalten. Alle drei Spalten sind rechtsbündig ausgerichtet. Die erste Spalte trägt die Überschrift **Zahl**, hat eine Breite von 4 Zeichen und enthält i . Die zweite Spalte trägt die Überschrift **Quadrat**, hat eine Breite von 7 Zeichen und enthält i^2 . Die dritte Spalte trägt die Überschrift **Kehrwert**, hat eine Breite von 8 Zeichen und enthält $1/i$ auf genau 6 Nachkommastellen gerundet. Die Spalten sind durch genau ein Leerzeichen getrennt.

Kommentieren Sie ihr Programm gründlich und verwenden Sie eine Schleife!

► **Aufgabe 1.2** [10 Punkte, [Link zur Abgabe](#)] Das Programm `error.c` enthält einige Syntaxfehler. Sorgen Sie dafür, dass das Programm mit `gcc -Wall` ohne Warnungen kompiliert indem Sie alle Syntaxfehler beheben. Lassen Sie die Ausgabe des Programmes unverändert!

Markieren Sie Ihre Änderungen mit Kommentaren.

Sie bekommen hier ausnahmsweise Teilpunkte, auch wenn Ihr Programm nicht kompiliert.

► **Aufgabe 1.3** [10 Punkte, [Link zur Abgabe](#)] Schreiben Sie ein Programm, das Zahlen von der Konsole einliest und mit InsertionSort sortiert.

Ihr Programm erhält die Eingabe als Text auf der Standardeingabe `stdin`. Die Eingabe enthält zunächst eine Zeile mit einer Zahl n . Es folgen n ganze Zahlen a_1, \dots, a_n in je einer Zeile. Geben Sie die Zahlen a_1, \dots, a_n in aufsteigend sortierter Reihenfolge getrennt durch ein Leerzeichen auf der Standardausgabe `stdout` aus.

Sie dürfen davon ausgehen, dass die Eingabe das spezifizierte Format hat (genau $n + 1$ Zeilen mit je genau einer ganzen Zahl). Es ist also keine Fehlerüberprüfung der Eingabe notwendig.

Hinweis: Das Handout enthält einen (aktualisierten) Abschnitt über das Einlesen von Zahlen!