

Übungsblatt 7: Programmieren in C (WS 2018/19)

Abgabe: Montag, 10.12.18, 12:00

1. Dieses Übungsblatt ist ein **Pflichtübungsblatt**, d.h. Sie müssen bei der Bearbeitung dieses Blattes **mind. 50% der Gesamtpunkte** erreichen.
2. Die Abgabe erfolgt über das Exclaim-System gemeinsam mit Ihrem Teampartner.
3. Zur Beantwortung von Fragen und Hilfestellung bei der Bearbeitung kommen Sie bitte in die wöchentliche Fragestunde, Mittwoch 15:30, in Terminalraum 32-410.
4. Bitte laden Sie einzelne Dateien hoch, wie in der Aufgabenstellung angegeben, keine Archive, keine kompilierten Dateien.
5. Programme, die nicht kompilieren, werden nicht korrigiert und mit **0 Punkten** bewertet.
6. Bitte beachten Sie unsere Regelung bei Plagiaten:
 - Wenn Sie sekundäre Quellen, wie Bücher oder das Internet verwenden, müssen Sie immer die Quelle angeben. Das einfache Kopieren aus anderen Quellen ist für die Übungen nicht gestattet. Wenn wir in einer Übungsabgabe kopierten Code ohne Quellenangabe finden, wird die gesamte Abgabe mit 0 Punkten bewertet.
 - Sie können Übungsaufgaben gerne mit den Mitgliedern anderer Teams diskutieren. Sie sollten jedoch Ihren Code vor der Abgabefrist nicht an andere Teams weitergeben bzw. zeigen.
 - Wenn Code von anderen Teams kopiert wurde, werden die Abgaben **von beiden Teams** mit 0 Punkten bewertet.
 - Wir behalten uns vor Punkte auch nachträglich abzuziehen, wenn ein Verstoß erst später bemerkt wird.

Aufgabe 1: Schaltjahr (2 Punkte)

Abgabe: `schaltjahr.c`

In unserem Kalender sind zum Ausgleich der astronomischen und kalendarischen Jahreslänge in regelmäßigen Abständen Schaltjahre eingebaut. Zur exakten Festlegung der Schaltjahre dienen die folgenden Regeln:

- Ist die Jahreszahl durch 4 teilbar, so ist das Jahr ein Schaltjahr.

Diese Regel hat allerdings eine Ausnahme:

- Ist die Jahreszahl durch 100 teilbar, so ist das Jahr kein Schaltjahr.

Diese Ausnahme hat wiederum eine Ausnahme:

- Ist die Jahreszahl durch 400 teilbar, so ist das Jahr doch ein Schaltjahr.

Erstellen Sie ein Programm, das berechnet, ob eine vom Benutzer eingegebene Jahreszahl ein Schaltjahr bezeichnet oder nicht.

Beispiel: Bei Eingabe von 2018 soll `Kein Schaltjahr`, bei Eingabe von 2020 soll `Schaltjahr` ausgegeben werden.

Aufgabe 2: Häufigkeit von Buchstaben (6 Punkte)

Abgabe: `buchstaben.c` in Exclaim

Schreiben Sie ein C-Programm, das für einen eingegebenen String die Häufigkeit einzelner Buchstaben ermittelt und **in einem Array abgespeichert werden**. Der String verwendet ausschließlich Großbuchstaben und umfasst maximal 60 Zeichen.

Um einen String auf sichere Art einzulesen, verwenden wir die Funktion `fgets(char * str, int n, FILE * stream)`, die `n` Zeichen aus dem einem Eingabestrom in den Speicherbereich schreibt,

der durch `str` referenziert wird. Als Eingabestrom verwenden wir hier `stdin`, d.h. die Kommandozeile.

```
char eingabe[N];
fgets(eingabe, N, stdin);
```

Dabei soll `N` den Wert 61 haben, um genügend Speicherplatz für max. 60 Zeichen plus dem Stringterminalsymbol `\0` bereitzuhalten.

Beispiel

Eingabe:

Ausgabe:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A: 1
B: 1
C: 1
D: 1
E: 1
F: 1
G: 1
H: 1
I: 1
J: 1
K: 1
L: 1
M: 1
N: 1
O: 1
P: 1
Q: 1
R: 1
S: 1
T: 1
U: 1
V: 1
W: 1
X: 1
Y: 1
Z: 1

Aufgabe 3: Das Tagebuch meines Bruders (8 Punkte)

Abgabe: `tagebuch.c`

Um Daten sicher zu kommunizieren, verwendet man heutzutage asymmetrische Verschlüsselungsalgorithmen wie RSA. Mein Bruder verwendet einen wesentlich einfacheren Algorithmus, um seine Tagebucheinträge vor fremden Blicken zu schützen. Er nutzt dazu einen Ersetzungsalgorithmus bei dem jeder Buchstabe im Ursprungstext durch einen anderen Buchstaben aus dem Alphabet ersetzt wird. Der Abstand zwischen den Buchstaben im Ursprungstext und dem verschlüsselten Text ist dabei konstant. Wenn man den konstanten Abstand d mit 5 definiert, wird der Buchstabe A durch F, B durch G, C durch H, ..., Y durch D, Z durch E ersetzt.

Mit einem bekannten konstanten Abstand d wäre es ziemlich einfach die Verschlüsselung zu knacken. Mein Bruder verwendet daher einen zufälligen Abstand für jeden neuen Eintrag. Um die Einträge zu entschlüsseln, muss ich daher für jeden Eintrag den Abstand d neu ermitteln. Dabei nutzte ich folgende Taktik: Der Buchstabe E wird mit Abstand am häufigsten in deutschen Texten verwendet. Allerdings ist es ziemlich mühsam, dies von Hand zu ermitteln.

Schreiben Sie ein C-Programm, das für einen eingegebenen Text den Abstand d ermittelt und den Text entschlüsselt. Der Text verwendet ausschließlich Großbuchstaben und Leerzeichen und umfasst maximal 60 Zeichen. Leerzeichen werden nicht verschlüsselt. Falls eine Entschlüsselung nicht möglich ist, da es mehrere Distanzen gibt, die verwendet werden könnten, geben Sie aus: `Entschluesselung nicht moeglich` Bei Eingabe von `NFJO UBHFCVDI` soll der Text `MEIN TAGEBUCH`

ausgegeben werden.

Hinweise:

1. Verwenden Sie Ihre Lösung aus Aufgabe 3!
2. Bei der Eingabe von Strings auf der Kommandozeile wird bei Verwendung von `fgets` der Zeilenumbruch als Zeichen dem Eingabestring hinzugefügt. Dies zusätzliche Zeichen im umgewandelten Text können Sie beim Testen ignorieren.¹

¹Alternativ können Sie Ihren Eingabestring in einer Textdatei abspeichern und diese bei der Ausführung auf Kommandozeile folgendermaßen als Eingabe nutzen:

```
gcc tagebuch.c  
./a.out < eingabe.txt
```