

Hinweise zu den Testaten und deren Bewertung

VERSIONSGESCHICHTE

NUMMER	DATUM	BESCHREIBUNG	NAME
v5.0.2	2020-05-08 10:35:04 +0200	Adresse zum Online Compiler angepasst	STI

Inhaltsverzeichnis

1	Allgemeines	1
2	Die Plagiats-Regelung	1
3	Automatische Korrektur	1
3.1	Lassen Sie keine „DEBUG“ Ausgaben in Ihrer Lösung stehen!	1
3.2	Reichern Sie <code>scanf</code> Befehle nicht durch vorangestellt <code>printf</code> 's an!	2
3.3	Reichern Sie auch <code>printf</code> Ausgaben nicht durch zusätzlichen Text an!	2
3.4	Reichern Sie Ihr Programm nicht mit Befehlen an um den Ablauf des Programms zu pausieren!	2
3.5	Allgemein gilt für Ein- und Ausgaben	3
4	Hinweise zur Benennung	3
5	Check vor der Abgabe (gcc.php)	3
6	Testen Sie Ihr Programm	4
7	Die Abgabe	4
8	Noch Fragen?	4

1 Allgemeines

Alle Hinweise stehen auf dem Aufgabenblatt zu Testat 0.

AUSZUG DER WICHTIGSTEN PUNKTE

- Sie müssen **sechs** der neun Testate bestanden haben um zur Prüfung zugelassen zu werden.
- Abzugeben ist genau ein C-Programm, wie in der jeweiligen Aufgabe beschrieben. (siehe Hinweise zur Benennung)
- Alle Programme müssen unter Linux-GCC fehler- und warnungsfrei übersetzbar sein, und zwar mit der Warnstufe (`gcc -Wall`). (siehe `gcc.php`)

2 Die Plagiats-Regelung



Wichtig

Die Plagiats-Regelung für aktive (abschreiben) und passive (abschreiben lassen) Plagiate gilt und wird angewendet! **Diese Regelung gilt für das Gesamt-Testat, d.h. ein Plagiat in einer einzigen der neuen Aufgaben führt zum Ausschluss / Nichtbestehen der 5 Kreditpunkte!**

Ihre Lösungen werden mit Programmen zur Auffindung von Plagiaten untersucht.

ES WERDEN U.A. FOLGENDE VERSCHLEIERUNGSVERSUCHE ERKANNT:

- Umbenennen von Variablen und Funktionen.
- Änderungen in den Einrückungen
- Vertauschen von Codeblöcken.

Verwenden Sie Ihre Kreativität lieber auf das eigenständige lösen der Aufgabe statt auf das Verschleiern eines Plagiats. Am Ende erwischen wir Sie trotzdem.

3 Automatische Korrektur

- Aufgrund der großen Anzahl von Studenten in diesem Kurs werden Ihre Lösungen automatisch korrigiert.
- Stichprobenartig werden einige Abgaben per Hand nachkontrolliert.
- Es wird die Logdatei des automatischen Korrekturlaufs bei mir gespeichert.

3.1 Lassen Sie keine „DEBUG“ Ausgaben in Ihrer Lösung stehen!

So nicht:

```
...
while(i < 5) {
    printf("Schleifendurchlauf Nr. %d\n", i);
    ....
}
```

3.2 Reichern Sie `scanf` Befehle nicht durch vorangestellt `printf`'s an!

Heißt es in der Aufgabenstellung: „Das Programm liest die Jahreszahl ein (genau ein `scanf`)“

```
scanf("%d", ...)
```

Dann darf folgendes NICHT im Code stehen:

```
...
printf("Geben Sie die Jahreszahl ein: ");
scanf("%d", &Jahr);
...
```

sondern nur die `scanf` Zeile.

3.3 Reichern Sie auch `printf` Ausgaben nicht durch zusätzlichen Text an!

Heißt es in der Aufgabenstellung: „Das Programm soll im Fehlerfall die Meldung "Fehler" (mit Zeilenumbruch) ausgeben“

Dann darf folgendes NICHT im Code stehen:

```
printf("Es ist ein Fehler aufgetreten.\n"); // zusätzlicher Text
printf("Fehler!\n");                       // zusätzliche Satzzeichen
printf("Fehler");                          // fehlender Zeilenumbruch
```

sondern wirklich nur:

```
printf("Fehler\n");
```

3.4 Reichern Sie Ihr Programm nicht mit Befehlen an um den Ablauf des Programms zu pausieren!

FOLGENDE AUFRUFE HABEN IN IHREM PROGRAMM NICHTS ZU SUCHEN:

- `getchar()`
- `system("pause")`
- ...

Tipp

Arbeiten Sie unter Windows mit einer IDE, so kann es sein, dass beim Drücken der "Ausführen" Knopfes Ihr Programm kurz in einer Konsole geöffnet wird die sich gleich wieder schließt.

In diesem Fall fügen Sie keinen Pausecode in Ihr Programm ein, sondern öffnen eine Konsole, wechseln in das Verzeichnis in dem die kompilierte EXE-Datei liegt und rufen diese einfach auf.

3.5 Allgemein gilt für Ein- und Ausgaben

Steht im Text „genau ein“ `scanf` oder `printf` so ist damit gemeint, dass **im kompletten Programm** genau ein `scanf` bzw. `printf` vorkommen darf.

In der Regel wird in der Aufgabe sogar der Formatstring mit angegeben. In diesem Fall **verwenden** Sie diesen genauso wie er da steht.

Steht im Aufgabentext, dass Ihr Programm seine Eingabe über die **Kommandozeile** bekommt, dann wird zum Einlesen der Eingabe **kein** `scanf` verwendet, sondern die Eingaben beim Programmaufruf mitgegeben:

```
<myprogramm> Eingabe1 Eingabe2 ...
```

4 Hinweise zur Benennung

Die Dateiergung Ihrer Quelldatei muss `.c` lauten (nicht `.cpp`).

IDEs (und auch `gcc` selbst) erkennen an der Endung welchen Übersetzer Sie verwenden müssen; den für `C` oder den für `C++`. Haben Sie versehentlich Sprachelemente von `C++` verwendet, so erkennt der Compiler — bei richtig benannter Datei — dass Sie einen Fehler gemacht haben und meldet dies.

Um die Lösungen richtig zuzuordnen muss der Name diesem Schema folgen:

```
<matrikel-nr>-testat-<x>.c
```

Ist Ihre Matrikelnummer `12345` und das Testat hat die Nummer `0`, so muss Ihre Datei wie folgt heißen:

```
12345-testat-0.c
```

Beachten Sie die Bindestriche und die Kleinschreibung!

5 Check vor der Abgabe (gcc.php)

Um zu prüfen, ob das Programm sich fehler- und warnungsfrei übersetzen lässt, haben wir die folgende Webseite zur Verfügung gestellt:

<https://image.informatik.htw-aalen.de/~stigler/gcc-web-aufruf>

WAS PRÜFT DIESE SEITE:

- Lässt sich Ihre Lösung als C-Programm übersetzen.
- Werden dabei Warnungen oder Fehler angezeigt (`gcc -Wall`).
- Entspricht der Name Ihrer Datei dem eben vorgestellten Schema.

WAS PRÜFT DIESE SEITE NICHT:

- Läuft das übersetzte Programm.
- Stellt Ihr Programm die eine Lösung der Aufgabe dar.
- Stimmt Ihre Matrikelnummer.

6 Testen Sie Ihr Programm

Nachdem Sie über die Webseite geprüft haben, dass Ihr Programm fehler- und warnungsfrei übersetzt werden kann, testen Sie ob es auch das tut, was es laut Aufgabenstellung machen soll!

Prüfen Sie Ihr Programm mit **verschieden** Eingaben!

Lassen Sie ruhig auch einen Ihrer Kommilitonen Ihr Programm testen.

7 Die Abgabe

- Die Abgabe erfolgt über den Canvas Kurs (<https://aalen.instructure.com> → Programmieren 1) direkt in der Testat-Aufgabenstellung.
- Achten Sie darauf, dass Sie den **Quelltext** Ihrer Lösung abgeben, nicht das kompilierte Programm, den Header oder die Projekt-Datei Ihrer IDE.
- Benennung Sie Ihre Datei gemäß dem Namensschema, falls Sie dies nicht schon gemacht haben.
- Beachten Sie die Abgabezeit auf dem Aufgabenblatt. Sobald diese Zeit abgelaufen ist, ist eine Abgabe nicht mehr möglich!
- Geben Sie frühzeitig ab! Technische Probleme treten meist dann auf, wenn es eilt!

8 Noch Fragen?

???