

3. Übung am 25.10.2010

=====
Anleitung:

Während jeder Übung ist von jeder Gruppe ein (kurzes) Protokoll zu erstellen. Das Protokoll ist eine einfache ASCII-Text-Datei, die mit einem Text-Editor (z.B. gedit, kate) mit dem Sie auch Ihre Programme schreiben, erzeugt wird. Nennen Sie diese Datei unbedingt "PROTOKOLL.txt".

Das Protokoll muss Folgendes enthalten:

1. Datum, Übungsnummer, Gruppennummer, Name(n) der mitwirkenden Studierenden,
2. Benötigter Zeitaufwand für die gestellten Aufgaben (circa),
3. Namen der erstellten Programme (KEINE Listings),
4. Kurze Antwort auf eventuell weiter unten gestellte Fragen,
5. Eventuelle Probleme oder Besonderheiten, falls diese aufgetreten sind.

Sämtliche während der Übung erstellten Dateien (Protokoll, Source Codes, ausführbare Programme, etc.) verbleiben im Verzeichnis für den jeweiligen Übungstag, also z.B. "03Ue2010-10-25/" Ihrer Gruppe.

Das Protokoll und die Übungsprogramme sollten am jeweiligen Übungsnachmittag erstellt werden, spätestens jedoch bis zum nächstfolgenden Übungstag (Montag), 14:00! (Spätere Ausarbeitungen können nur in begründeten Fällen und nach Rücksprache mit Ihrem Betreuer bzw. Tutor berücksichtigt werden!)

=====

Aufgaben zu Kapitel 4 (3 Punkte):

1. Schreiben Sie ein C-Programm, das alle Primzahlen bis zu einer wählbaren maximalen Zahl ausgibt. Ein Verfahren dafür ist das "Sieb von Eratosthenes". Dieser Algorithmus verwendet ein Feld `f[i]`, wobei `f[i]=1` gilt falls `i` eine Primzahl ist, sonst ist der Wert 0.

Das erreicht man, indem man für jedes `i` alle Elemente, die einem beliebigen Vielfachen von `i` entsprechen, auf 0 setzt. Eine Zahl, die ein Vielfaches einer anderen Zahl ist, kann keine Primzahl sein.

Für die Ausgabe (am Bildschirm) muss das Feld nochmals durchlaufen werden.

Fragen:

1. Was ist in folgendem C-Programmteil vermutlich falsch?

```
#include <stdio.h>

void main()
{int i=0;
  if(i=5)printf("%d ist gleich 5\n",i);
  else printf("%d ist nicht gleich 5\n",i);
}
```

Falls Sie die Lösung nicht selbst finden, kompilieren Sie das Programm mit folgendem Schalter:

```
gcc -Wall prog.c
```

Was fällt Ihnen dabei auf?

2. Wie potenziert man eine Zahl in C?

3. `{int i=0;}`

Welchen Wert liefert `{printf("%i", ++i);}`

bzw. `{printf("%i", i++);}` ?