

## Prozedurale Programmierung

Präsenzaufgaben Termin VI: Einfache Algorithmen, binäre Suche, Zufallszahlen.

1. Schreiben Sie eine Funktion, welche alle Primzahlen zwischen 2 und 100 bestimmt und ausgibt. Gehe dafür folgendermaßen vor: Zuerst wird ein Array mit den Zahlen  $2, 3, \dots, 100$  initialisiert. Beginnend mit der 2 werden alle Vielfachen der zuletzt gefundenen Primzahl markiert (gleich 0 gesetzt). Die neue Primzahl ist die nächste unmarkierte Zahl. Ist das Quadrat der zuletzt gefundenen Primzahl größer als 100, so enthält das Array bereits die gesuchten Primzahlen.
2. Schreiben Sie ein vollständiges Programm mit zwei Funktionen `swap1` und `swap2`, die zwei übergebene Eingabeargumente (`int`) vertauschen. In Funktion `swap1` sollen die Parameter durch *Call By Value* und in `swap2` durch *Call By Reference* übergeben werden. Erklären Sie den Unterschied zwischen *Call By Value* und *Call By Reference*.
3. Schreiben Sie ein Programm, das sechs wöchentliche Lottozahlen (6 aus 49) vorschlägt. Beachten Sie dabei, dass beim Lotto keine Zahl doppelt vorkommen darf. Verwenden Sie dafür den Zufallszahlengenerator aus der Standardbibliothek `<stdlib.h>`.

**Tip:** `rand()` liefert eine Pseudo-Zufallszahl aus dem Bereich von 0 bis `RAND_MAX`. Zur Projektion der Zufallszahl auf ein Intervall `[a,b]` berechne den ganzzahligen Rest bei Division durch `b-a+1` und addiere `a` dazu.

4. Schreibe ein Programm, das dem Benutzer folgendes Auswahlmenü anzeigt:

(1) <code>a + b</code>	(2) <code>a - b</code>	(5) Programmende
(3) <code>a * b</code>	(4) <code>a / b</code>	

Das Programm soll die Tastatureingabe einer Ziffer zwischen 1 und 5 erwarten und eine entsprechende Operation auf zwei einzulesende `int`-Zahlen anwenden. Eine Fehlerbehandlung bei Division durch 0 wird erwartet.

5. Schreiben Sie ein vollständiges Programm, das eine ganze Zahl  $n > 0$  von der Tastatur einliest und anschließend zwei `double`-Arrays `v,w` der Länge  $n$  erzeugt und diese ebenfalls von der Tastatur einliest. In einer Funktion `dotprod` soll das Skalarprodukt beider Arrays  $s = \sum_{k=0}^{n-1} v[k] \cdot w[k]$  berechnet und im Hauptprogramm ausgegeben werden.
6. Schreiben Sie ein Makro `mymac`, das 100 mal "Hello world" ausgibt. Die einzige Befehlszeile der `main`-Funktion sei der Makroaufruf `mymac`.
7. Makros können auch Parameter enthalten:

```
#define makroname(param1,param2,...) makrobefehl(e)
```

Schreiben Sie ein Makro `max`, welches das Maximum zweier Zahlen berechnet.