

Praktikum Programmieren  
Übungsblatt 6  
Termin: 23. November 2023

### Übung 1 „Tauschen von Werten“

Vervollständigen Sie das folgende Code-Fragment so, dass am Ende die Werte von `m` und `n` vertauscht sind:

```
1 #include <stdio.h>
2
3 int main(){
4     int m = 9;
5     int n = 10;
6     int* pm;
7     int* pn;
8     ... // ???
9     printf("m = %i , n = %i .", m, n);
10    return 0;
11 }
```

Die Variablen `m` und `n` dürfen dabei nicht direkt Ziele von Zuweisungen sein! Verwenden Sie stattdessen Zeiger, um den Zugriff auf die Variablen zu realisieren. Sie dürfen eine Hilfsvariable vom Typ `int` verwenden.

### Übung 2 „Arrays ohne [] – die Erste“

Schreiben Sie ein Programm, das einen String einliest, dessen Länge ermittelt und diese ausgibt. Nutzen Sie für die Ermittlung der Länge *keine* Bibliotheksfunktionen (wie z.B. `strlen`) und verwenden Sie dabei nicht den Array-Index-Operator `[]`!

### Übung 3 „Arrays ohne [] – die Zweite“

Schreiben Sie ein Programm, das ein Array mit 50 Einträgen vom Typ `int` mit den ersten 50 Fibonacci-Zahlen initialisieren. Geben Sie dann den Inhalt des Arrays mit Hilfe von `printf` aus. Verwenden Sie dabei nicht den Array-Index-Operator `[]`!

Zur Erinnerung: Die Fibonacci-Folge  $f_{(k)}$  ist für  $k \in \mathbb{N}$  wie folgt definiert:

$$f_{(k)} = \begin{cases} 1 & \text{für } k \in \{1, 2\} \\ f_{(k-2)} + f_{(k-1)} & \text{für } k > 2 \end{cases}$$

Orientieren Sie sich für die Ausgabe an folgenden Format:

Fibonacci-Zahlen:

```
f(1) = 1
f(2) = 1
f(3) = 2
f(4) = 3
f(5) = 5
```

f(6) = 8  
...

#### Übung 4 „Quadratfunktion“

Schreiben Sie eine Funktion mit dem Namen `hoch2`, die für eine als Parameter übergebene ganze Zahl deren Quadrat berechnet und zurückgibt.

Testen Sie die Funktion wie folgt:

```
1 ...  
2 for (int i = 0; i < 20; i++) {  
3     printf("hoch2(%i) = %i\n", i, hoch2(i));  
4 }  
5 ...
```

#### Übung 5 „String umdrehen“

Schreiben Sie eine Funktion `void dreh(const char* s)`, die die Zeichenkette `s` in umgedrehter Reihenfolge ausgibt. Testen Sie die Funktion mindestens mit einer leeren Zeichenkette, mit einer Zeichenkette der Länge 1 und weiteren Zeichenketten länger als 1.

#### Übung 6 „Zweierpotenzen“

Schreiben Sie eine Funktion `void zp(int a, int b)`, die für zwei gegebene Zahlen `a` und `b` alle Zweierpotenzen im Intervall  $[a, b]$  ausgibt.

So soll der Aufruf `zp(10, 200)` die folgende Ausgabe erzeugen: 16 32 64 128.

Fangen Sie unsinnige Eingaben mit entsprechende `if`-Anweisungen ab und geben Sie in diesen Fällen eine Fehlermeldung aus.

#### Übung 7 „Teilstrings zählen“

Schreiben Sie eine Funktion `int vorkommen(const char* s, const char* m)`, die zählt, wie oft das Muster `m` im String `s` vorkommt. Die Funktion soll diese Anzahl als Rückgabewert zurückgeben.

Beispielsweise soll beim Aufruf `vorkommen("Dies ist ein Beispiel", "ie")` der Wert 2 zurückgegeben werden.

Nutzen Sie für die Zählung nur selbstgeschriebenen Code.