OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG
INFORMATIK UND MATHEMATIK

# Übungsaufgaben Blatt 5 - Spaß mit Pointern

Programmieren 1 - Einführung in die Programmierung mit C - Prof. Dr. Ruben Jubeh

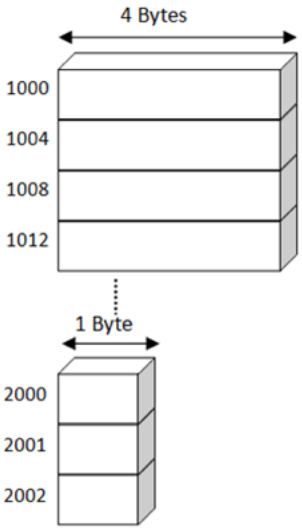
---

## Aufgabe 1 : Pointer

Gegeben sind folgende Anweisungen:

```
1 char c = 'T', d = 'S';
2 char *p1 = &c;
3 char *p2 = &d;
4 char *p3;
```

1. Beschriften Sie das vorgegebene Diagramm so, dass die Zuordnung von Variablen zu Speicherzellen und die Speicherbelegung klar ersichtlich sind. Wir nehmen dabei an, dass ein `char` Wert **1 Byte** groß ist, und ein Pointer immer **4 Byte** belegt.




2. Nun werden folgende Anweisungen ausgeführt:

```
1 p3 = &d;
2 printf("%c", *p3);
```

3. Nun werden folgende Anweisungen ausgeführt:

```
p3 = p1;
printf("%c", *p3);
```

Was wird ausgegeben?

4. Nun werden folgende Anweisungen ausgeführt:

```
*p1 = *p2;
printf("%c", *p1);
```

Was wird ausgegeben?

**Aufgabe 2** : **StarWars in C**

Das folgende Beispiel kombiniert Parameterübergabe, *Call by Value*, *Call by Reference* und Pointer.
Gehen Sie den Programmcode Schritt für Schritt durch, zeichnen Sie dabei die Repräsentation der Variablen im Speicher und geben Sie an, welche Ausgaben das Programm erzeugt. Lösen Sie diese Aufgabe ohne Computer.

```c
/*
* File: StarWarsInC.c
* This program explores parameters, return values, scalar types and pointers.
 */

#include "simpio.h"

void lukeSkywalker(int chewbacca, int *toto) {
    chewbacca = 100;
    *toto = 44;
}

int darthVader(int *emperor, int r2d2) {
    emperor = &r2d2;
    r2d2 = 14;
    return *emperor;
}

int landoCalrissian(int *alderaan, int hanSolo) {
    hanSolo = 'S';
    alderaan = &hanSolo;
    return *alderaan;
}

int main() {
    char hanSolo = 'h';
    int alderaanValue = 10;
    int *alderaan = &alderaanValue;
    int r2d2 = 137;

    int emperorValue = 30;
    int *emperor = &emperorValue;

    lukeSkywalker(*alderaan, &emperorValue);
    printf("Alderaan: %d\n", *alderaan);
    printf("Emperor: %d\n", *emperor);

    darthVader(emperor, r2d2);
    printf("r2d2: %d\n", r2d2);
    printf("Emperor: %d\n", *emperor);

    *alderaan = landoCalrissian(alderaan, hanSolo);
    printf("hanSolo: %c\n", hanSolo);
    printf("Alderaan %c\n", *alderaan);
    return 0;
}
```

**Aufgabe 3** : **Der Header *stdio.h***

Kompilieren Sie das folgenden Codebeispiel und führen es aus:

```c
#include <stdio.h>

int main() {
  int number;

  printf("Enter a number:");
  scanf("%d", &number);
  printf("You entered: %d", number);
  return 0;
}
```

Beantworten Sie anschließend die folgenden Fragen:

- Was macht das Programm?
- Finden Sie heraus (Internetrecherche), welche Funktionalität der Header `stdio.h` Ihnen als Programmierer zur Verfügung stellt. Was ist aus Ihrer Sicht der Unterschied zum Header `simpio.h`?
- Erläutern Sie die Funktion `scanf` mit Bezug zu den Konzepten *Call by value* und *Call by reference*.

**Aufgabe 4** : **Call-by-Reference**

Gegeben ist folgendes Programm:

```c
int cube(int n) {
  return n * n * n;
}

int main() {
  int number = 5;
  printf("The number: %d\n", number);
  printf("The result: %d\n", cube(number));
  return 0;
}
```

Schreiben Sie die Funktion so um, dass Sie mit *Call-By-Reference* arbeitet. Die neue Signatur der Funktion ist `void cube(int *n)`. Passen Sie auch die main-Funktion an. Für Ihre Lösung brauchen Sie kein `return` mehr.

**Aufgabe 5** : **Refactoring des BMI-Rechners von *call by value* auf *call by reference***

Bauen Sie Ihren BMI-Rechner so um, dass die Berechnung des BMI über *call by reference* anstatt *call by value* funktioniert. Verwenden Sie die Musterlösung aus Grips als Startpunkt, falls Sie die BMI-Aufgabe nicht selbst lösen konnten.