

Arbeitsblatt: INF1

Name:

Kurznamen:

Arrays, Zeiger, Datenstrukturen

Aufgabe 1: Einführung, Zeiger und Adressen

Legen Sie in der *main()*-Funktion zwei *int*-Variablen *a* und *b* an sowie zwei Zeiger *pa* und *pb*, die auf die beiden Variablen zeigen. Belegen Sie *a* und *b* mit Werten, indem Sie über die Zeiger auf die Variablen zuweisen.

Geben Sie die Adressen der beiden Variablen aus (in *printf* mit *%p*).

Addieren Sie die beiden Variablen, indem Sie über die Zeiger auf die Variablen zugreifen und geben Sie auch die Summe aus.

Die Ausgabe des Testprogramms könnte etwa so aussehen:

```
Die Variable a liegt auf der Adresse 0x7fff60538cb4
Die Variable b liegt auf der Adresse 0x7fff60538cb0
Summe: 42
```

Aufgabe 2: Arrays Lottozahlengenerator



Es soll ein Programm zur Erzeugung von Lottozahlen erstellt werden. Es sollen sechs (unterschiedliche) aus 42 Zahlen zufällig ausgewählt und angezeigt werden. Implementieren Sie die Funktion ***void lottozahlen(int* zahlen)***.

Hinweise:

- Zahlen in einen Array speichern.
- verwenden Sie um die Arraygrösse bzw. Anzahl Ziehungen festzulegen:

```
#define ZIEHUNGEN 6
```

```
#define MAXLOTTOZAHLEN 42
```
- Sie müssen sicherstellen, dass jede Zahl nur einmal gezogen wird, d.h. bei jeder neuen Zahl muss jeweils überprüft werden, ob die Zahl nicht schon gezogen wurde.
- In den Vorlesungsunterlagen finden Sie ein Programm das würfelt i.e. Zahlen zwischen 1 und 6 bestimmt.
- Es lässt sich übrigens zeigen, dass die mit einem Lottozahlengenerator erzeugten Zahlenfolgen bessere Gewinnchancen bieten als die von Hand erzeugten.

Abgabe

Praktikum: INF7.2

Filename: lottozahlen.c

Aufgabe 3: Anzahl Richtige

Die Ziehung am letzten Wochenende hat die Gewinnzahlen 4,8,9,18,27,34 ergeben. Schreiben Sie eine Funktion, die die *Anzahl richtiger Werte Ihrer Zufallszahlen* bestimmt in der Funktion `int richtige(int* zahlen, int* richtig)`

Abgabe

Praktikum: INF7.3

Filename: lottozahlen.c

Aufgabe 4: Verwendung von Pointern

Schreiben Sie obiges Programm so um, dass konsequent für die Arrayzugriffe Pointer Variablen verwendet werden. Implementieren Sie die beiden obigen Funktionen nochmals aber mit Pointern.

```
int zahlen[6];  
int *pzahlen = &(zahlen[0]);
```

Abgabe

Praktikum: INF7.4

Filename: lottozahlenPointer.c