

Einführung in die Informatik

Aufgabenblatt 8

- a) Schreiben Sie ein Programm mit zwei `int` Variablen `x` und `y`. Erstellen Sie dann einen Zeiger auf `x`. Schreiben Sie über den Zeiger die Zahl 5 in `x` und geben Sie anschließend `x` auf dem Bildschirm aus. Lassen Sie danach den Zeiger auf `y` zeigen und schreiben Sie die Zahl 20 über den Zeiger in `y` und geben Sie auch diesen Wert aus.
- b) Schreiben Sie ein einfaches Programm, das die folgenden Definitionen von Variablen und die geforderten Anweisungen enthält:
- Definition einer Variablen `i` vom Typ `int`
 - Definition eines Pointers `ptr` vom Typ `int *`
 - Zuweisung der Adresse von `i` an den Zeiger `ptr`
 - Zuweisung des Wertes 1 an die Variable `i`
 - Ausgabe des Wertes des Zeigers `ptr`
 - Ausgabe des Wertes von `i`
 - Ausgabe des Wertes des Objekts, auf das der Zeiger `ptr` zeigt, mit Hilfe des Dereferenzierungsoperators `*`,
 - Zuweisung des Wertes 2 an das Objekt, auf das der Zeiger `ptr` zeigt, mit Hilfe des Dereferenzierungsoperators `*`
 - und Ausgabe des Wertes von `i`

Hinweis: Zeiger werden bei `printf()` mit dem Platzhalter `%p` ausgegeben.

- c) Was ist hier nicht richtig?

```
#include <stdio.h>

int main()
{
    int x;
    int *p_x = 0;

    *p_x = &x;

    printf("Inhalt von x = %d", *p_x);

    return 0;
}
```

- d) Gehen Sie davon aus, dass Sie eine Variable `int x` und einen Zeiger `int *p_x` haben. Der Zeiger `p_x` zeigt auf die Variable `x`. Welcher Wert ergibt sich durch die folgenden Schreibweisen? Welche dieser Schreibweisen sind identisch? Testen Sie Ihre Aussagen anhand eines geeigneten Programms!

`x`
`&x`
`*x`

`p_x`
`&p_x`
`*p_x`

- e) Welche Ausgabe erzeugt das folgende Programmfragment? Bevor Sie es eingeben und ausprobieren, Sie überlegen, wie wohl die Programmausgabe aussehen wird.

```
int i = 9, *pi;

pi = &i;
printf("i = %d  &i = %u\n", i, &i);
printf("*pi = %d  pi = %u\n", *pi, pi);

*pi = 3;
printf("i = %d  &i = %u\n", i, &i);
printf("*pi = %d  pi = %u\n", *pi, pi);
printf("&pi = %u\n", &pi);
```

- f) Schreiben Sie eine C-Funktion

```
int teile(int i, int j, int *e, int *r);
```

welche die übergebene Ganzzahl `i` durch `j` dividiert und in `e` das Ergebnis sowie in `r` den Divisionsrest zurückliefert. Alle Zahlen sollen ganze Zahlen sein. Die Funktion `teile()` soll eine Division durch 0 abfangen und in diesem Fall den Rückgabewert 0 liefern, sonst den Rückgabewert 1.

Testen Sie Ihre Funktion mit einer geeigneten `main()`-Funktion aus. Zeigen Sie, wie das 3. und 4. Argument beim Funktionsaufruf als (1) Zeigervariable bzw. (2) Nicht-Zeigervariable realisiert werden kann – testen Sie diese 2 Varianten des Funktionsaufrufs.

g) Warum könnte es hier Probleme geben?

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int *numbers;
    int max;

    printf("Maximalzahl: ");
    fflush(stdin);
    scanf("%d", &max);

    numbers = malloc(max * sizeof(int));

    return 0;
}
```

h) Schreiben Sie ein Programm, das den Benutzer fragt, wie viele Zeichen er eingeben will. Daraufhin soll das Programm den Speicherplatz für die Zeichen reservieren und schließlich die Zeichen einlesen, die Größe des hierfür reservierten Speicherplatzes anzeigen und die Zeichen wieder ausgeben.

i) Warum funktioniert das Programm nicht?

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int *p_x = 0;

    p_x = (int *) malloc(sizeof(int));

    *p_x = 23;

    free(p_x);

    printf("Inhalt von x = %d\n", *p_x);

    return 0;
}
```