

Praktikum Programmieren  
Übungsblatt 5  
Termin: 16. November 2023

### Übung 1 „Zufallszahlen 1“

Legen Sie ein Array mit 20 Elementen des Typs `int` an und füllen Sie es mit Zufallszahlen. Verwenden Sie dazu die Funktion `rand()` von Übungsblatt 1. Um nicht immer dieselbe Zufallszahlen zu bekommen, fügen Sie zu Beginn Ihrer `main`-Funktion die Anweisung `srand(time(0));` ein.

Für die Funktion `rand()` und die Funktion `time()` müssen Sie die folgenden `#include`-Anweisungen ganz am Anfang Ihres Codes hinzufügen:

```
1 #include <stdlib.h>
2 #include <time.h>
```

Geben Sie das mit Zufallszahlen gefüllte Array aus.

Iterieren Sie dann über das mit Zufallszahlen gefüllte Array und addieren Sie die geraden und die ungeraden Zahlen getrennt auf. Deklarieren Sie dafür die Variablen `sum_even` für die Summe der geraden Zahlen und `sum_odd` für die Summe der ungeraden Zahlen. Geben Sie die Summen aus.

Lassen Sie das Programm mindestens zweimal laufen und überprüfen Sie die ausgegebenen Summen auf Korrektheit.

### Übung 2 „Zufallszahlen 2“

Schreiben Sie ein Programm, das zunächst ein Array mit 100 zufälligen Ganzzahlen (s. Aufgabe 1) im Bereich 0 bis 50 füllt. Füllen Sie dann ein Array mit 51 Einträgen, indem Sie an den Index `i` schreiben, wie oft die Zahl `i` im ersten Array vorkommt.

Geben Sie beide Arrays aus und prüfen Sie in mindestens drei verschiedenen Durchläufen für jeweils mindestens 5 Zahlen stichprobenmäßig, ob das Ergebnis korrekt ist.

### Übung 3 „Binärdarstellung“

Lesen Sie eine Ganzzahl ein und speichern Sie ihre Binärdarstellung in der „richtigen“ Reihenfolge in einem Ganzzahl-Array. Jedes Element in diesem Array soll entweder den Wert 0 oder 1 enthalten. Geben Sie das Array abschließend aus.

Prüfen Sie mit einigen Zahlen, ob die Binärdarstellung korrekt berechnet wird.

### Übung 4 „Länge einer Zeichenkette“

Schreiben Sie ein Programm, das eine Zeichenkette einliest und bestimmen Sie durch selbstgeschriebenen Code die Länge der Zeichenkette. Vergleichen Sie Ihr Ergebnis mit der Ausgabe von `strlen()`.

Testen Sie Ihr Programm mit mindestens einer leeren Eingabe, einer Eingabe mit nur einem Zeichen und einer Eingabe mit mehreren Zeichen.

### Übung 5 „Zeichenkettenvergleich“

Schreiben Sie ein Programm, das zwei Zeichenketten einliest und sie miteinander vergleicht. Geben Sie eine Meldung aus, aus der hervorgeht, ob die beiden Zeichenketten gleich sind.

### Übung 6 „Zeichenkette umdrehen“

Schreiben Sie ein Programm, das eine Zeichenkette einliest und in umgekehrter Reihenfolge abspeichert. Geben Sie die umgekehrte Zeichenkette aus.

Testen Sie das Programm mit mindestens einer leeren Eingabe, einer Eingabe mit einem Zeichen, einer Eingabe mit einer geraden Zahl von Zeichen und einer Eingabe mit einer ungeraden Zahl von Zeichen.

### Übung 7 „Zeichenketten ohne Vokale“

Schreiben Sie ein Programm, das eine Zeichenkette einliest und alle Vokale entfernt. Beispiel: Aus „Informatik“ soll „nfrmtk“ werden. Geben Sie die neue Zeichenkette aus.

Testen Sie das Programm mit mindestens einer leeren Eingabe, einer Eingabe bestehend nur aus Vokalen und einer Eingabe komplett ohne Vokale sowie einer „normalen“ Eingabe, die Vokale und Nicht-Vokale enthält.

### Übung 8 „Zeichenketten zusammenfügen“

Schreiben Sie ein Programm, das zwei Zeichenketten einliest und die zweite Zeichenkette an die erste anhängt. Geben Sie das Ergebnis.

Schreiben Sie zwei Varianten des Programms: Eine, die die Funktion `strcat` verwendet, und eine, in der Sie das Zusammenfügen selbst implementieren.

Testen Sie das Programm mit mindestens zwei leeren Eingaben, einem Fall, in dem nur die erste Eingabe leer ist, einem Fall, in dem nur die zweite Eingabe leer ist, und einem Fall, in dem beide Eingaben nicht leer sind.