

Matrikel-Nr.:

Prof. Dr. U. Schrader  
FRA-UAS / Fb 2

C-Programmierung  
Klausur am 01.03.2022

### Aufgabe 1 (6 Punkte)

Schreiben Sie eine Funktion, die einen String prüft, ob es sich dabei um ein Palindrom handelt. Palindrome sind Worte oder Sätze, die vorwärts oder rückwärts gelesen, dasselbe ergeben, z.B. "Reittier". Die Funktion soll dabei keinen Unterschied zwischen Groß- und Kleinschreibung machen. Testen Sie die Funktion mit den folgenden Strings: "Reittier", "Lagerregal", "Step on no pets", "Teststring". Geben Sie das Ergebnis des Tests am Bildschirm aus.

Speichern Sie Ihr Programm im Verzeichnis **Ergebnisse** in der Datei **aufgabe1.c**

---

*Write a function that checks a string, whether it is a palindrome. A palindrome is a word, phrase, or other sequence of characters which reads the same backward as forward, such as madam or racecar. The function shall make no difference between upper and lower case. Test the function with the strings: "Reittier", "Lagerregal", "Step on no pets", "Teststring". Display the results on the screen.*

*Store your program in the folder "Ergebnisse" as file "aufgabe1.c"*

### Aufgabe 2 (6 Punkte)

Schreiben Sie eine Funktion, die ein Array von Gleitkommazahlen übergeben bekommt und die größte Zahl, die kleinste Zahl und den Mittelwert aller Zahlen zurückgibt. Testen Sie Ihre Funktion mit einem Array der Länge 10, welches mit Zufallszahlen gefüllt wird. Geben Sie das Array und die drei Kennzahlen am Bildschirm aus.

Speichern Sie Ihr Programm im Verzeichnis **Ergebnisse** in der Datei **aufgabe2.c**

---

*Write a function that receives an array of float values, and returns the maximum, the minimum, and the mean of the values. Test your function with an array of length 10, that is filled with random numbers. Display the array and the three return values on the screen.*

*Store your program in the folder "Ergebnisse" as file "aufgabe2.c"*

### Aufgabe 3 (6 Punkte)

Schreiben Sie ein Programm, dass eine von Ihnen eingegebene Integerzahl in seine Primzahl faktoren zerlegt. Wird also beispielsweise die Zahl 84 eingegeben, soll als Ergebnis am Bildschirm das Folgende ausgegeben werden:

84 = 2\*2\*3\*7

Speichern Sie Ihr Programm im Verzeichnis **Ergebnisse** in der Datei **aufgabe3.c**

---

*Write a program, that factors a number that you have entered into primes. If you enter for example the number 84, the display on the screen should look like:*

84 = 2\*2\*3\*7

Matrikel-Nr.:

Prof. Dr. U. Schrader  
FRA-UAS / Fb 2

C-Programmierung  
Klausur am 01.03.2022

Store your program in the folder "Ergebnisse" as file "aufgabe3.c"

**Aufgabe 4 (6 Punkte)** ✓

Öffnen Sie die Datei "input.txt" im Verzeichnis "vorlagen" zum Lesen. Geben Sie den Inhalt der Datei auf dem Bildschirm aus.

Speichern Sie Ihr Programm im Verzeichnis Ergebnisse in der Datei aufgabe4.c

---

Store your program in the folder "Ergebnisse" as file "aufgabe4.c" *Write a program, that opens the file "input.txt" in the folder "vorlagen" for reading and displays its contents on the screen.*

*Store your program in the folder "Ergebnisse" as file "aufgabe4.c"*

**Aufgabe 5 (9 Punkte)** ✓

Schreiben Sie ein Programm, dass die Datei input.txt im Verzeichnis Vorlagen zum Lesen öffnet. Bestimmen Sie die Anzahl der Worte, die durchschnittliche Wortlänge in Zeichen, die Anzahl der Sätze und die Anzahl der Zeilen. Geben Sie diese Statistiken am Bildschirm aus.

Speichern Sie Ihr Programm im Verzeichnis Ergebnisse in der Datei aufgabe5.c

---

*Write a program, that opens the file "input.txt" in the folder "vorlagen" for reading. The program shall determine the average length of a word in the text, and the number of words, the number sentences, and lines. The statistics will be displayed on the screen. Store your program in the folder "Ergebnisse" as file "aufgabe5.c"*

**Aufgabe 6 (12 Punkte)**

Schreiben Sie ein Programm, dass dynamisch einen sogenannten Stack als verkettete Liste realisiert. Ein Stack hat im Wesentlichen nur zwei Funktionen: mit "push()" wird ein neues Element angelegt und am Anfang des Stacks gespeichert, mit "pop()" holt man den Inhalt des obersten Elements vom Stack. Danach wird dieses Element vom Stack entfernt und der dafür allokierte Speicher wird freigegeben. Ein Kartenstapel, auf dem man immer eine Karte ablegt oder von oben eine herunternimmt, ist ein solcher Stack. Das Last-In-First-Out Prinzip eines Stacks ist in der Abb. 1 erläutert. Sie können die Datei "linkedlist.c" im Ordner "Vorlagen" als Basis für Ihre Lösung nutzen. Das Programm soll nacheinander mittels "push()" die Spielkarten "Pik-As", "Pik-Koenig", "Pik-Dame", "Pik-Bube" und "Pik-10" auf dem zunächst leeren Stack ablegen, um diese dann mittels "pop()" nacheinander wieder vom Stack zu holen und dabei den Namen der Spielkarte am Bildschirm ausgeben.

Speichern Sie Ihr Programm im Verzeichnis Ergebnisse in der Datei aufgabe6.c