

Name: _____ Matrikelnummer: _____

PC-Nummer: _____ Unterschrift: _____

bearbeitete Aufgaben: ☐ 1 ☐ 2 ☐ 3 ☐ 4

von den Prüfenden auszufüllen:

<input type="checkbox"/> Bonus	<input type="checkbox"/> kein Bonus	<input type="checkbox"/> bestanden	<input type="checkbox"/> nicht bestanden
--------------------------------	-------------------------------------	------------------------------------	--

Leibniz Universität Hannover
Fachgebiet Mensch-Computer-Interaktion

Testklausur Programmieren 1
2. September 2022, XX:00 Uhr

Gruppe T

Dies ist eine **90**-minütige Klausur. Tragen Sie bitte Ihren Namen, Ihre Matrikelnummer, die PC-Nummer und die bearbeiteten Aufgaben auf diesem Blatt ein und unterschreiben Sie es. Legen Sie bitte Ihren Studierendenausweis und Ihren Personalausweis zur Anwesenheitskontrolle bereit. Es sind keine zusätzlichen Hilfsmittel erlaubt. Schalten Sie bitte Ihr Mobiltelefon aus und legen Sie es weg. Für Notizen können Sie diesen Zettel verwenden. Notizen auf diesem Zettel werden nicht mitbewertet.

Wenn Sie **keinen Bonus** erlangt haben, müssen Sie von den vier gestellten Aufgaben **drei Aufgaben** erfolgreich lösen. **Mit Bonus** müssen Sie nur **zwei** der vier Aufgaben erfolgreich lösen.

Hinweise:

- Loggen Sie sich ein mit Benutzername `uni` und Passwort `uni`.
- Öffnen Sie das Terminal über die Menüs Anwendungen / Systemwerkzeuge / Terminal und wechseln Sie mit `cd` in das Template-Verzeichnis `Home / Schreibtisch / KlausurT / groupT`. Listen Sie die Dateien mit `ls`.
- Öffnen Sie das Template-Verzeichnis mit dem Dateimanager per Doppelklick. Öffnen Sie KWrite über Anwendungen / Zubehör / KWrite.
- Aktivieren Sie Zeilennummern in KWrite mit `F11`.
- Hinweise zum Kompilieren auf der Kommandozeile finden Sie in den Template-Dateien.
- Ihre Lösungen müssen mit den jeweiligen Testfällen funktionieren und auch generelle Lösungen darstellen. Die Testfunktionen und Testfälle dürfen nicht verändert werden.
- Sie dürfen beliebige Hilfsfunktionen implementieren.

1. Die Template-Datei für diese Aufgabe ist `four_sorted_digits.c`. Implementieren Sie die Funktion `bool four_sorted_digits(String s)`. Diese Funktion soll `true` zurückgeben, wenn `s` mindestens vier hintereinander stehende und aufsteigend sortierte Dezimalziffern enthält. Sonst soll die Funktion `false` zurückgeben. Wenn Vorbedingungen hilfreich sind, fügen Sie mit diese `require("Beschreibung", Bedingung);` hinzu.

Dokumentieren Sie Ihre Lösung mit einem Purpose-Statement und mit Quelltext-Kommentaren!

Hinweis: Zur Erinnerung: `String` ist definiert als `typedef char* String`. Ein `String` ist ein `char`-Array mit terminierendem 0-Zeichen. Daher können Indizes verwendet werden, um auf die einzelnen Buchstaben zuzugreifen: `s[i]`.

2. Die Template-Datei für diese Aufgabe ist `center_or_zero.c`. Implementieren Sie die Funktion `center_or_zero`. Diese Funktion soll die Zahl in der Mitte der Liste zurückgeben. Wenn die Liste eine gerade Anzahl an Elementen enthält, soll die Funktion 0 zurückgeben.

Dokumentieren Sie Ihre Lösung mit einem Purpose-Statement und mit Quelltext-Kommentaren!

3. Die Template-Datei für diese Aufgabe ist `rotate_array.c`. Implementieren Sie die Funktion `void rotate(int* a, int n, int r)`, die das `int`-Array `a` der Länge `n` Elemente um `r` Stellen rotiert. Wenn `r > 0`, wird nach links rotiert. Wenn `r < 0`, wird nach rechts rotiert. Zum Beispiel wird aus `a = [1, 2, 3, 4, 5]` mittels `rotate(a, 5, 2)` das Array `[3, 4, 5, 1, 2]`. Das Array wurde also um 2 Stellen nach links rotiert.

Dokumentieren Sie Ihre Lösung mit einem Purpose-Statement und mit Quelltext-Kommentaren!

Hinweis: Ein temporäres Array kann in `rotate` auf dem Stack mittels `int b[n];` angelegt werden.

4. Die Template-Datei für diese Aufgabe ist `nodes_equal_to_parent.c`. Implementieren Sie die Funktion `int number_of_nodes_that_are_equal_to_their_parent(Node* node)`. Diese Funktion soll die Anzahl der Knoten des Binärbaums zurückgeben, die den gleichen Wert haben, wie ihre Elternknoten. Wenn Vorbedingungen hilfreich sind, fügen Sie diese mit diese `require("Beschreibung", Bedingung);` hinzu.

Dokumentieren Sie Ihre Lösung mit einem Purpose-Statement und mit Quelltext-Kommentaren!