Aufgabe 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 4 hintereinander stehende und aufsteigend sortierte Dezimalziffern enthält. Sonst gibt die Funktion false zurück. Wenn Vorbedingungen hilfreich sind, fügen Sie diese 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].

Aufgabe 2.

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 hinzu.

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

Aufgabe 3.

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. Die Template-Datei für diese Aufgabe ist center\_or\_zero.c

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

Aufgabe 4.

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.