

Präsenzaufgabe 5.1

Programmieren Sie in Java einen Algorithmus, der die Werte eines Integerarrays in umgekehrter Reihenfolge in ein neues Array speichert.

Aufgabe 5.1 (5 Punkte).

Schreiben Sie ein Javaprogramm, das alle Einträge eines Integerarrays daraufhin überprüft, ob zwei von ihnen in Summe 42 ergeben. Alle gefundenen Pärchen sollen auf der Konsole ausgegeben werden.

Aufgabe 5.2 (8 Punkte).

Programmieren Sie einen Algorithmus, der eine Matrizenmultiplikation durchführen kann. Die beiden Matrizen und auch die Ergebnismatrix werden durch zweidimensionale Integerarrays dargestellt. Sie können davon ausgehen, dass die Arraydimensionen > 0 sind und die Arrays valide Matrizen abbilden. Allerdings wissen Sie nicht, ob die Matrizen für die Multiplikation passende Dimensionen besitzen.

Geben Sie außerdem die Ergebnismatrix auf der Konsole aus. Ein anschauliches Beispiel einer Matrizenmultiplikation finden Sie unter:

<https://de.wikipedia.org/wiki/Matrizenmultiplikation>.

Aufgabe 5.3 (8 Punkte).

Eine Aufgabe aus der Klausur des letzten Semesters.

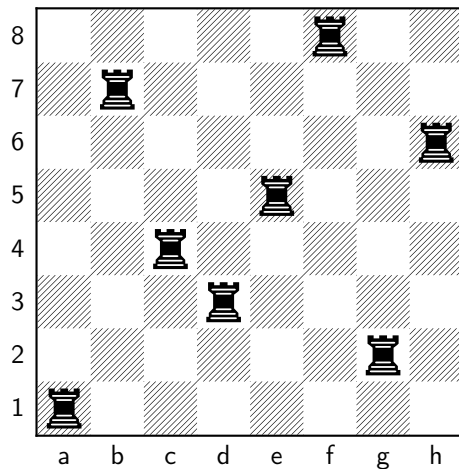
In dieser Aufgabe beschäftigen wir uns mit dem n -Türme Problem:

Auf einem $n \times n$ Felder großen Spielfeld sollen n Türme so positioniert werden, dass sie sich nicht gegenseitig schlagen können.

Türme können nur gerade nach oben, unten, rechts oder links ziehen. Dabei können sie beliebig viele Felder weit ziehen, aber nur in eine Richtung.

Ein Turm schlägt einen anderen Turm, wenn er auf dessen Feld zieht.

Folgende Abbildung zeigt eine korrekte Lösung für das n -Türme Problem:



Das Spielfeld wird mit einem $n \times n$ großen boolean Array dargestellt, wobei jeder Eintrag des Arrays ein Feld repräsentiert. Ein Eintrag ist `true` wenn auf dem entsprechenden Feld ein Turm positioniert ist.

Schreiben Sie ein Programm, das überprüft ob n Türme korrekt aufgestellt wurden. Das Programm soll dann entsprechend auf der Konsole ausgeben, ob die Positionierung korrekt oder falsch ist.