Einschränkungen

- Sie dürfen **keine** zusätzlichen eigenen Hilfsmethoden oder globalen Variablen verwenden.
- Die vorgegebenen Methodenköpfe dürfen **nicht** erweitert oder geändert werden.
- Für die Implementierung der rekursiven Methode dürfen **keine** Schleifen verwendet werden.
- Sie dürfen Strings **nicht** per Referenz vergleichen.
- Sie dürfen **nicht** die Methoden `clone` und `System.arraycopy` verwenden.
- Sie dürfen **nur** folgende Methode(n) aus der Klasse `Arrays` verwenden: `deepToString`, `toString`
- Sie dürfen **nur** folgende Methode(n) aus der Klasse `String` verwenden: `charAt`, `equals`, `length`, `substring`, `isEmpty`

Aufgabenstellung

Deklarieren und initialisieren Sie in `main` die folgende(n) Variable(n):

```
int[][] test1 = {{0, 2, 4}, {2, 0, 0}, {0, 0, 1}};
int[][] test2 = {{1, 2, 3}, {1, 2, 3, 4, 5}, {1, 2, 3}, {1, 2, 3, 4, 5}};
int[][] test3 = {{2}, {0, 7}, {6, 7, 8}, {6, 0}, {0, 0}};
String seq1 = "ABA";
```

Implementieren Sie folgende Methoden:

- **`int[][] removeLeadingZeros(int[][] inputArray)`** erzeugt aus `inputArray` ein neues zweidimensionales Array und retourniert dieses. Das neue Array übernimmt jede Zeile aus dem `inputArray` und entfernt führende Nullen.
  Vorbedingung(en): `inputArray.length > 0`.
  Wird die Methode z.B. mit `test1` aufgerufen, entsteht folgendes Array:

2	4	
2	0	0
1		

- **`void mask(int[][] inputArray, int[] rows, int[] cols)`** setzt alle Werte in `inputArray` auf 0, wenn deren Zeilenindex in `rows` **und** deren Spaltenindex in `cols` vorkommt.
  Vorbedingung(en): `inputArray.length > 0`, die Werte in `rows` und `cols` sind `>=0` und aufsteigend sortiert.
  Wird die Methode z.B. mit `inputArray=test2`, `rows={1,2,3}` und `cols={0,1,4}` aufgerufen, entsteht folgendes Array:

1	2	3		
0	0	3	4	0
0	0	3		
0	0	3	4	0

- **`String replicateCharacters(String sequence, String repSequence)`** gibt einen neuen String zurück, bei dem das i-te Zeichen im String `sequence` zusätzlich wiederholt wird, wenn der i-te Eintrag im String `repSequence` eine 1 ist.
  Diese Methode muss rekursiv implementiert werden.
  Vorbedingung(en): `sequence != null`, `repSequence != null`
  `sequence.length() == repSequence.length()`, alle Zeichen in `repSequence` sind Ziffern im Bereich 0 bis 1.

Deklarieren Sie auch neue Arrays, die für die Tests benötigt werden.
Testen Sie alle Methoden und deren Seiteneffekte in `main` mit zumindest folgenden Aufrufen und weiteren Aufrufen (z.B. mit `deepToString`) für die Ausgaben.

Aufruf	**Ausgabe in main auf der Konsole**
`result1 = removeLeadingZeros(test1)`	[[2, 4], [2, 0, 0], [1]]
`result2 = removeLeadingZeros(test3)`	[[2], [7], [6, 7, 8], [6, 0], []]
`mask(test2,new int[]{1,2,3},new int[]{0,1,4})`	[[1, 2, 3], [0, 0, 3, 4, 0], [0, 0, 3], [0, 0, 3, 4, 0]]
`mask(test3,new int[]{0,2,4},new int[]{0,1})`	[[0], [0, 7], [0, 0, 8], [6, 0], [0, 0]]
`mask(test1,new int[]{},new int[]{0,1})`	[[0, 2, 4], [2, 0, 0], [0, 0, 1]]
`replicateCharacters(seq1, "010")`	ABBA
`replicateCharacters("SAMBA", "10001")`	SSAMBAA

Methode	**Bewertungsgrundlage**	**Punkt(e)**
`main`	Deklarationen	/ 1
	Testfälle korrekt implementiert	/ 2
`removeLeadingZeros`	Richtige Dimensionen	/ 1
	Korrekte Schleifen	/ 2
	Korrekte Längen aller Zeilen	/ 3
	Korrekter Inhalt in den Zeilen	/ 3
`mask`	Korrekte Schleifen	/ 4
	Korrekte Elemente auf 0 gesetzt	/ 5
`replicateCharacters`	Korrekter Methodenansatz (Rückgabe vorhanden)	/ 1
	Basisfall vorhanden	/ 1
	Basisfall korrekt	/ 1
	Forschritt der Rekursion vorhanden	/ 1
	Forschritt der Rekursion korrekt	/ 1
	Korrekter Rückgabewert	/ 4
Gesamt		/ 30