# Practice - 2D arrays - drawing the data

## P1. Assume you have a method `buildArray(N)` creates and returns an int[] of size N, with values 1,2,...,N:

```java
public static int[] buildArray(int N)
```

The following code is executed:

```java
int[][] table;
int[] arr = null;
table = new int[3][];
table[1] = buildArray[5];
table[2] = new int[7];
table[0] = table[1];
table[1] = arr;
```

Draw the JAVA picture for each line of code (i.e. show actual Java implementation). Redraw the program state (all variables and all memory in use) for each new line. Read the instructions by column.

<table>
<thead>
<tr>
<th>int[][] table;</th>
<th>table[2] = new int[7];</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>int[] arr = null;</td>
<td>table[0] = table[1];</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>table = new int[3][];</td>
<td>table[1] = arr;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>table[1] = buildArray[5]; // Is the above line valid?</td>
<td></td>
</tr>
</tbody>
</table>
P2. Write a method that takes as argument a 2D matrix and returns the transposed matrix. E.g. for:

\[
\begin{array}{ccc}
  16 & 4 & 9 \\
  3 & 7 & 11 \\
\end{array}
\]

returns

\[
\begin{array}{ccc}
  16 & 3 \\
  4 & 7 \\
  9 & 11 \\
\end{array}
\]

Draw the data after each main step. If nested loops, draw the data after each iteration of the OUTER loop. – Note you have to do a similar drawing for hw 8.

Problem solving steps:

- Identify smaller components for solving this problem.
- Think about the data: draw what type of data the method is given and what it returns. Next think about how you can produce that.