3.2.6 An ***order n matrix*** is an $n \times n$ array of numbers. For example,

$$(3)$$

is a 1×1 matrix,

$$\begin{matrix} 1 & 3 \\ -2 & 8 \end{matrix}$$

is a 2×2 matrix and

$$\begin{matrix} 1 & 3 & 4 & 6 \\ 2 & -5 & 0 & 8 \\ 3 & 7 & 6 & 4 \\ 2 & 0 & 9 & -1 \end{matrix}$$

is a 4×4 matrix. Define the ***minor*** of an element x in a matrix as the submatrix formed by deleting the row and column containing x. In the above example of a 4×4 matrix, the minor of the element 7 is the 3×3 matrix

$$\begin{matrix} 1 & 4 & 6 \\ 2 & 0 & 8 \\ 2 & 9 & -1 \end{matrix}$$

Clearly, the order of a minor of any element is 1 less than the order of the original matrix. Denote the minor of an element $a[i, j]$ by $minor(a[i, j])$.
Define the ***determinant*** of a matrix a (written $det(a)$) recursively as follows:

1. If a is a 1×1 matrix (x), then $det(a) = x$.
2. If a is of an order greater than 1, compute the determinant of a as follows:

   a. Choose any row or column. For each element $a[i, j]$ in this row or column, form the product:

   ```
   power(-1,i + j) * a[i, j] * det(minor(a[i, j]))
   ```

   where i and j are the row and column positions of the element chosen, $a[i, j]$ is the element chosen, $det(minor(a[I, j]))$ is the determinant of the minor of $a[i, j]$, and $power(m, n)$ is the value of m raised to the nth power.

b. $det(a)$ = sum of all these products.
(More concisely, if n is the order of a,

```
det(a) = Σi power(-1, i + j) * a[i, j] * det(minor(a[i,
j])), for any j
```

or

```
det(a) = Σj power(-1, i + j) * a[i, j] * det(minor(a[i,
j])), for any i).
```

Write a Java program that reads a, prints a in matrix form, and prints the value of $det(a)$, where det is a method that computes the determinant of a matrix.