Question #1 Solution

```
public static int[][] swapRowsAndColumns(int[][] square)
{
   int[][] newSquare = new int[square.length][square[0].length];

   for (int row = 0; row < square.length; row++)
   {
      for (int col = 0; col < square[0].length; col++)
      {
         newSquare[row][col] = square[col][row];
      }
   }

   return newSquare;
}
```

Question #2, Solution #1

```
public static int borderSum(int[][] matrix)
{
   int sum = 0;

   // Iterate through every element in the matrix, only adding numbers along the border
   for (int row = 0; row < matrix.length; row++)
      for (int col = 0; col < matrix[0].length; col++)
         if ((row == 0) || (col == 0) || (row == matrix.length - 1) || (col == matrix[0].length - 1))
            sum += matrix[row][col];

   return sum;
}
```

Question #2, Solution #2

```
public static int borderSum(int[][] matrix)
{
   int sum = 0;

   // Iterate across the top and bottom rows of the matrix
   for (int col = 0; col < matrix[0].length; col++)
      sum += matrix[0][col] + matrix[matrix.length - 1][col];

   // Iterate down the left and right columns of the matrix, skipping the corners
   for (int row = 1; row < matrix.length - 1; row++)
      sum += matrix[row][0] + matrix[row][matrix[0].length - 1];

   return sum;
}
```