For the following problems you may NOT use any Matlab bultin function (unless it is explicitly stated) or language structure (such as `for`, `if`, etc.)

1. Write a one-line Matlab command that yields 1 if a 2D matrix, A, has ***at least one*** complex entry (i.e. $x + y$`i`), and 0 otherwise.

2. Write an anonymous, one-line, Matlab function (`stirling = @(X)..`) that computes the Stirling's approximation for factorials. Your function must work for either a scalar or any multidimensional matrix X. If the input is a matrix, then your function must approximate the factorial of each of its elements. The Stirling's approximation to $n!$ is given by

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

   where $e = 2.71828\ldots$

3. The *Pascal* matrix, S, is a $p \times p$ square matrix containing the binomial coefficientes as its elements, i.e.

$$S_{ij} = \left(\begin{array}{c} n \\ r \end{array}\right) = \frac{n!}{r!(n-r)!}$$

   where $n = i + j$ and $r = i$. In other words:

$$S_{ij} = {}_{i+j}\mathrm{C}_i = \frac{(i+j)!}{(i)!(j)!}$$

   where i and j are the row and column ***zero-based*** indexes respectively, e.g., $0 \leqslant i \leqslant p-1$ and $0 \leqslant j \leqslant p-1$.
   Write a Matlab function that receives a single, scalar number p, and generates the $p \times p$ square Pascal matrix. (*Hint: Use your* `stirling` *function to approximate factorials in your computation*).