1. (Pratap 3.6.3) Write a function

\[
\text{function } [D,E,F] = \text{createspecialarrays}(M,N)
\]

which creates three arrays using \texttt{zeros}, \texttt{eye}, and \texttt{ones}. \(D\) should be \(M \times N\) (\(M\) rows, \(N\) columns) filled with 0’s, \(E\) should be \(M \times M\) filled with 3’s, and \(F\) should be \(N \times N\) with 5’s on the diagonal. For instance, if \(M = 2\) and \(N = 3\), you would have the arrays,

\[
D = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \quad E = \begin{bmatrix}
3 & 3 \\
3 & 3 \\
\end{bmatrix}, \quad F = \begin{bmatrix}
5 & 0 & 0 \\
0 & 5 & 0 \\
0 & 0 & 5 \\
\end{bmatrix}.
\]

Then write a script which calls your function \texttt{createspecialarrays} for \(M = 3\) and \(N = 2\) and displays the outputs \(D\), \(E\), and \(F\).

Note that in general \texttt{zeros} is a good way to initialize an array to make sure that memory is allocated efficiently and also (in particular for single-index arrays) that the array has the intended shape (or size). For a single-index column array of length \(m\) you should use \texttt{zeros(m,1)}; for a single-index row array of length \(n\) you should use \texttt{zeros(1,n)}.

2. Write a script which performs the following operations (in sequence)

(i) creates a \(20 \times 40\) array, \(A\), in which each element (or entry) in rows 1 through 10 is assigned the value 1 and each element in rows 11 through 20 is assigned the value 2;

(ii) creates a new \(20 \times 40\) array, \(B\), which is the same as \(A\) except row 11 for which \(B(11,j) = 1/j\), for \(1 \leq j \leq 40\);

(iii) creates a new \(20 \times 41\) array, \(C\), which is the same as \(B\) for columns 1 through 40 but also includes a column 41 in which all elements are assigned the value 3;

(iv) creates a new \(20 \times 41\) array, \(P\), which is the same as \(C\) except the first ten entries on the main diagonal for which \(P(i,i) = i \times C(i,i)\), for \(1 \leq i \leq 10\);

(v) creates a new \(20 \times 41\) array, \(Q\), which is the same as \(P\) except the (1,2) entry for which \(Q(1,2)\) is assigned the value 7;
(vi) creates a new $20 \times 41$ array, $R$, in which each element is the square of the corresponding element in $Q$ — for example, $R(1,2)$ will be assigned the value 49;

(vii) creates a scalar $\text{bigsum}$ which is the sum of all the elements (820 in total) of the array $R$.

You should use a judicious combination of MATLAB built-in functions, concatenation, for loops, and dotted operators.