1. Let the function $f$ be defined for positive integers by

$$f(n) = \begin{cases} 
3 & \text{if } n = 1 \\
0 & \text{if } n = 2 \\
2f(n-1) + f(n-2) & \text{otherwise}
\end{cases}$$

Write the function $f$ in Mathematica and use it to compute $f(20)$. Show all work.
2. Define a function \texttt{avg[]} that computes the average (mean) value of its arguments; the function should work for any number of arguments \( \geq 1 \), regardless of whether the arguments are given in a list or as a sequence. Examples:

\begin{verbatim}
In[]: avg[6,7]
Out[]: 6.5

In[]: avg[{6,7}]
Out[]: 6.5

In[]: avg[6,7,8,9]
Out[]: 8.5

In[]: avg[{6}]
Out[]: 6
\end{verbatim}

3. Use recursion to write a function \texttt{mult[m_,n_]} that, for integers \( m \leq n \), gives the product \( m(m+1)(m+2)\cdots n \). For example, \texttt{mult[2,4]} should give 24, since \( 2 \cdot 3 \cdot 4 = 24 \), and \texttt{mult[5,5]} should give 5. Your function should work for negative integers as well. You may not use the built-in Mathematica function \texttt{Factorial[]}.
4. Use recursion to write a function `fitin[lis_, n_]` that inserts `n` into a sorted list of numbers in the appropriate place, as in the following examples. Assume that `lis` is sorted in *descending order*, i.e., from large to small. You may not use the built-in Mathematica function `Sort[]`.

   In[]: fitin[{8, 7, 0, -2}, 3]
   Out[]: {8, 7, 3, 0, -2}

   In[]: fitin[{8, 7, 0, -2}, 7]
   Out[]: {8, 7, 7, 0, -2}

5. Use recursion to write a function `sortNumbers[lis_]` that sorts any nonempty list `lis` of numbers in *descending order*. Even if you didn’t do the previous problem (correctly), you may for this problem assume that the function `fitin[]` is given to you and use it. You may not use the built-in Mathematica function `Sort[]`. Example:

   In[]: sortNumbers[{4, 6, 0, -9, 3, 2, 6, 2}]
   Out[]: {6, 6, 4, 3, 2, 2, 0, -9}