Lecture 28: Recursion
**Factorial**

```python
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact(n-1)
```

![Diagram showing the computation of factorial for n=4]
Write a recursive version of exponentiation called \( \text{exp}(n, k) \) that computes \( n^k \). Note that exponentiation is repeated multiplication.

\[ \begin{align*}
\text{>>> } \text{exp}(2,0) & \quad 1 \\
\text{>>> } \text{exp}(2,1) & \quad 2 \\
\text{>>> } \text{exp}(2,2) & \quad 4 \\
\text{>>> } \text{exp}(2,3) & \quad 8 \\
\text{>>> } \text{exp}(2,10) & \quad 1024
\end{align*} \]
Write a recursive version of production called `prod(L)` that computes the product of the numbers in the list `L`.

```python
>>> prod(list(range(1,5)))
24
>>> prod(list(range(1,6)))
120
>>> prod(list(range(1,7)))
720
```
Write a recursive version of reverse called \texttt{rev(L)} that returns the members of \texttt{L} in reverse order.

\begin{verbatim}
>>> rev(list(range(10)))
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
>>> rev([1])
[1]
>>> rev([])
[]
\end{verbatim}