PS1 Solution Template

a. Using the matrix squaring operator create a "triangular" matrix with 1 on the main diagonal, 2 above, etc.

$$M(n) = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ & 1 & 2 & \dots & n-1 \\ & & \ddots & \ddots & \\ & & & 1 & 2 \\ & & & & 1 \end{pmatrix}$$

In[1]:
```
# The line below is a Julia function that given n computes M(n)
# If you are not using Julia please supply your function
M(n) = triu(ones(n,n))^2
# Show that it works
M(5)
```

Out[1]:
```
# Put your output here
```

In[2]:
```
# Show that it works for n=1,2,3,4,5,6
{M(n) for n=1:6}
```

Out[2]:
```
# Put your output here
```

b. You very likely have heard of the triangular numbers: (see wikipedia if not)

$$T_n = 1 + 2 + \ldots + n.$$

In[3]:
```
#Here they are
cumsum(1:10)'
# If not using Julia, how would you do this in your language?
```

Out[3]:
```
1x10 Array{Int64,2}:
 1  3  6  10  15  21  28  36  45  55
```

Don't use cumsum, or sum or "+", just matrix operations to create the matrix that has the triangular numbers on the diagonals: Explain roughly (not too formal a proof), why your idea works.

$$M(n) = \begin{pmatrix} 1 & 3 & \ldots & (n-1)n/2 & n(n+1)/2 \\ & 1 & 3 & \ldots & (n-1)n/2 \\ & & \ddots & \ddots & \\ & & & 1 & 3 \\ & & & & 1 \end{pmatrix}$$

In[4]:
```
# Put your input code here, put in a function and show that it
works, just like In[1] and Out[1]
```

Out[4]:

c. Don't stop. Keep going, and get the tetrahedral numbers. (see wikipedia) Explain briefly why this worked.

In[5]:
```
# This is hardly much different from In[4] and Out[4]
```

Out[5]:

d. Let $A = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}$ and $B = \begin{bmatrix} -1 & 2 & 1 & 4 \end{bmatrix}$.

Compute $(AB)^{10}$ on your computer. Explain why it's possible to get this without a computer!

```
In[6]:  A=[1;2;3;1]
```

```
Out[6]:  4-element Array{Int64,1}:
         1
         2
         3
         1
```

```
In[7]:  B=[-1 2 1 4]
```

```
Out[7]:  1x4 Array{Int64,2}:
         -1  2  1  4
```

```
In[8]:  (A*B)^10
```

```
Out[8]:  # Put your solution here
```

```
In[9]:  B*A
```

```
Out[9]:  # Put your solution here
```

and see if you can see and tell us what is going on