```
from numpy.polynomial import Polynomial as P
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return x *(x>0) - x*(x<0)

def g(n):
    #get x, fx
    xdata = [ -1 + 2.0*k/n  for k in range(n+1)]
    ydata = [ f(p) for p in xdata ]
    # use nevilles method
    fcns = [ P(y) for y in ydata]
    # initialise j, increment each time to indicate gij = p_
i,i+1,...i+j
    j = 0
    # keep combining until we get one function
    while (len(fcns) > 1):
        j = j + 1
        newfcns = []
        for i in range(len(fcns)-1):
            #method to combine 2 polys into one, to interpolate
1 extra point
            a = xdata[i]
            b = xdata[i+j]
            p1 = P([-a,1])
            p2 = P([-b,1])
            combined = (p1*fcns[i+1] - p2*fcns[i] ) / (b-a)
            newfcns.append(combined)
        fcns = newfcns
    return fcns[0]



#a:
x = np.linspace(-1,1,100)
y = [f(p) for p in x]
```

```
fig, axes = plt.subplots( nrows=3, ncols=1 )
axes[0].scatter(x, y)
axes[0].set_title("f(x)")

#b:
g2 = g(2)
g3 = g(3)
g4 = g(4)
g5 = g(5)
y2 = [g2(p) for p in x]
y3 = [g3(p) for p in x]
y4 = [g4(p) for p in x]
y5 = [g5(p) for p in x]
axes[1].plot(x, y, "r.",  x, y2, "g.", x, y3, "b.", x, y4, "y.",
x, y5, "c.")
axes[1].set_title("f,g2,g3,g4,g5 in colors
red,green,blue,yellow,cyan")

#c:
x = range(1, 21)
seq = [ (g(i))(0.3) for i in x]
axes[2].plot(x, seq, "r.")
axes[2].set_title("Plot of sequence gn(0.3),n=1 to 20")
plt.tight_layout()
fig.savefig('./151Hw2p4a.png')
plt.close(fig)
```