```
from numpy.polynomial import Polynomial as P
import numpy as np

def findPolynomialInterpolation(pts):
    xdata = [ pts[i][0] for i in range(len(pts)) ]
    ydata = [ pts[i][1] for i in range(len(pts)) ]
    # 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]

points = []
a = 0
loop = True
while (loop):
    inp = raw_input("Enter value 'a' when done, otherwise enter
a point in the form: x,f(x)\n")
    vals = inp.split(',')
    if(len(vals) == 1):
        loop = False
```

```
        a = float(vals[0])
    elif (len(vals) == 2):
        points.append([float(vals[0]), float(vals[1])])
    else:
        print("Try again. Example Valid inputs:
\n5,123\n22,61\n53")

output = findPolynomialInterpolation(points)(a)
print("Interpolated the points: " + str(points) + "\nAnd
evaluated at: " + str(a) +"\nWe have output: " + str(output))
```