EX.3.1.17, Sauer3

The estimated mean atmospheric concentration of carbon dioxide in earth's atmosphere is given in the table that follows, in parts per million by volume. Find the degree 3 interpolating polynomial of the data and use it to estimate the CO2 concentration in (a) 1950 and (b) 2050. (The actual concentration in 1950 was 310 ppm.)

year	CO_2 (ppm)
1800	280
1850	283
1900	291
2000	370

Please use COLAB for this problem.
No handwritten work! Please turn in a Colab Notebook

EX.3.1.17, Sauer3, solution, Langou

Colab: https://colab.research.google.com/drive/18ib4wsRwTsUqVwA6LxgIReG2Dc18pAO2

```
import copy
import numpy as np
import matplotlib.pyplot as plt


def newtdd_inplace( x, y ):
    n = len( x )
    for i in range(1,n):
        for j in range(n-i-1,-1,-1):
            y[j+i] = ( y[j+i] - y[j+i-1] ) / ( x[j+i] - x[j] )
    return y

def newtdd( x, y ):
    c = copy.deepcopy( y )
    newtdd_inplace( x, c )
    return c

def polyval_nested_w_base_points( c, b, x ):
  d = np.size(c)
  px = c[ d-1 ] * np.ones( np.shape(x) )
  for i in range( d-2, -1, -1 ):
    px = px * ( x - b[i] ) + c[i]
  return px
```

```
x = np.array([1800., 1850., 1900., 2000.])
y = np.array([ 280.,  283.,  291.,  370.])
```

```
c = newtdd( x, y )
```

```
print("coefficients of interpolating polynomial in nested form")
print("from degree 0 to higest degree:\n",c)

yy = polyval_nested_w_base_points( c, x, x )
err = abs( yy - y )
print("--------------------------------------------------------------- -----")
print("|    x     |           y            |           p(x)         |  error
|")
print("---------------------------------------------------------------------")
for i in range(0,len(x)):
    print("|",f"{x[i]: 8.1f}","|", f"{y[i]: 20.16f}","|", f"{yy[i]: 20.16f}","|"
print("---------------------------------------------------------------------")

print( "absolute error = ",  f"{np.linalg.norm( y - yy, np.infty):6.1e}" )
```

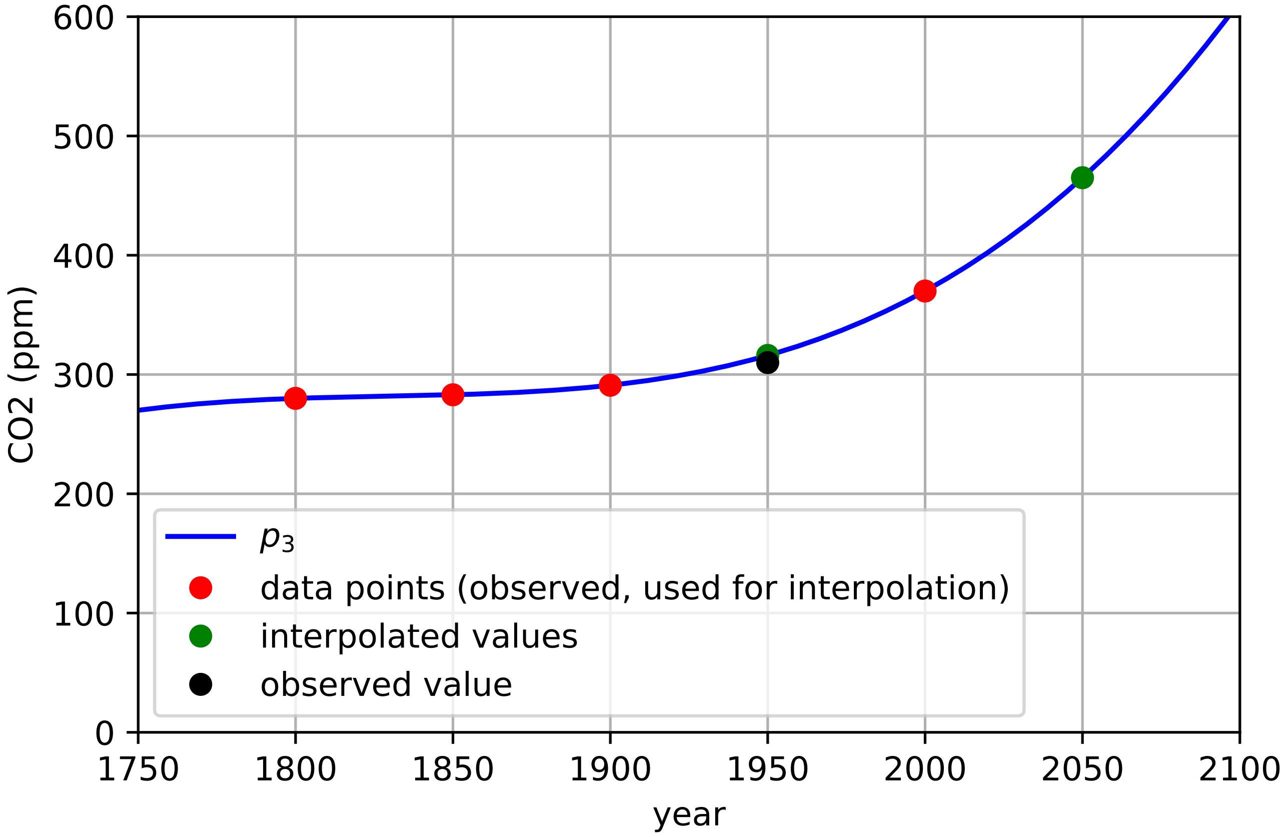
```
coefficients of interpolating polynomial in nested form
from degree 0 to higest degree:
 [2.8e+02 6.0e-02 1.0e-03 1.6e-05]
--------------------------------------------------------------- -----
|     x     |          y            |          p(x)          |  error    |
----------------------------------------------------------------------
|    1800.0 |  280.0000000000000000 |  280.0000000000000000 |   0.0e+00 |
|    1850.0 |  283.0000000000000000 |  283.0000000000000000 |   0.0e+00 |
|    1900.0 |  291.0000000000000000 |  291.0000000000000000 |   0.0e+00 |
|    2000.0 |  370.0000000000000000 |  370.0000000000000000 |   0.0e+00 |
----------------------------------------------------------------------
absolute error =  0.0e+00
```

```
print( "p(1950) = ",  f"{polyval_nested_w_base_points( c, x, 1950 ):8.2f}" )
print( "p(2050) = ",  f"{polyval_nested_w_base_points( c, x, 2050 ):8.2f}" )
```

```
p(1950) =    316.00
p(2050) =    465.00
```

```
### !!! What is below is NOT needed !!! ###
```

```
xx = np.linspace( 1750, 2100, 1000)
yy = polyval_nested_w_base_points( c, x, xx )
plt.plot(xx, yy, '-b',label=r'$p_3$')
plt.plot(x, y, 'or', label='data points (observed, used for interpolation)')
plt.plot([1950,2050], polyval_nested_w_base_points( c, x, [1950,2050] ), 'og', la
plt.plot(1950, 310, 'ok', label='observed value')
plt.xlabel('year')
plt.ylabel('CO2 (ppm)')
plt.xlim([ 1750, 2100 ])
plt.ylim([ 0., 600.])
plt.legend()
plt.grid()
plt.show()
```


600
500
400
300
200
100
0
CO2 (ppm)
1750
1800
1850
1900
1950
2000
2050
2100
year
p_3
data points (observed, used for interpolation)
interpolated values
observed value