Section 3.2

```
def neville(datax, datay, x):
      #datax: list of x value
      #datay: list of f(x) value
      #x: the x value used for approximating interpolation
    n = len(datax)
    p = n*[0]
    for k in range(n):
        for i in range(n-k):
            if k == 0:
                p[i] = datay[i]
            else:
                p[i]=((x-datax[i+k])*p[i]-(x-datax[i])*p[i+1])/(datax[i]-datax[i+k]
    return p[0]
```

Question 1a

```
#degree one
#p_0,1
x1 = [8.1, 8.3]
y1 = [16.94410, 17.56492]
neville(x1, y1, 8.4)
```

```
17.875329999999998
```

```
#p_1,2
x2 = [8.3, 8.6]
y2 = [17.56492, 18.50515]
neville(x2, y2, 8.4)
```

```
17.878330000000002
```

```
#p_2,3
x3 = [8.6, 8.7]
y3 = [18.50515, 18.82091]
neville(x3, y3, 8.4)
```

```
17.87363
```

```
#degree two
#p_0,1,2
x4 = [8.1, 8.3, 8.6]
y4 = [16.94410, 17.56492, 18.50515]
neville(x4, y4, 8.4)
```

```
17.87713
```

```
#p_1,2,3
x5 = [8.3, 8.6, 8.7]
y5 = [17.56492, 18.50515, 18.82091]
neville(x5, y5, 8.4)
```

```
17.877155000000002
```

```
#degree three
#p_0,1,2,3
x6 = [8.1, 8.3, 8.6, 8.7]
y6 = [16.94410, 17.56492, 18.50515, 18.82091]
neville(x6, y6, 8.4)
```

```
17.877142500000005
```

Question 1b

```
#degree one
#p_0,1
x7 = [-0.75, -0.5]
y7 = [-0.07181250, -0.02475]
neville(x7, y7, -1/3)
```

```
0.006625000000000006
```

```
#p_1,2
x8 = [-0.5, -0.25]
y8 = [-0.02475, 0.33493750]
neville(x8, y8, -1/3)
```

```
0.2150416666666667
```

```
#p_2,3
x9 = [-0.25, 0]
y9 = [0.33493750, 1.101]
neville(x9, y9, -1/3)
```

```
0.0795833333333334
```

```
#degree two
#p_0,1,2
x10 = [-0.75, -0.5, -0.25]
y10 = [-0.07181250, -0.02475, 0.33493750]
neville(x10, y10, -1/3)
```

```
0.18030555555555558
```

```
#p 1,2,3
```

```
x11 = [-0.5, -0.25, 0]
y11 = [-0.02475000, 0.33493750, 1.10100000]
neville(x11, y11, -1/3)
```

```
0.16988888888888892
```

```
#degree three
#p_0,1,2,3
x12 = [-0.75, -0.5, -0.25, 0]
y12 = [-0.07181250, -0.02475000, 0.33493750, 1.10100000]
neville(x12, y12, -1/3)
```

```
0.17451851851851855
```

Question 3

```
#a
x = [-2, -1, 0, 1, 2]
y = [1/9, 1/3, 1, 3, 9]
neville(x,y,1/2)
```

```
1.7083333333333335
```

```
#b
import math
x = [0,1,2,4,5]
y = [0,1, math.pow(2,1/2), 2, math.pow(5,1/2)]
neville(x,y,3)
```

```
1.6906067646231164
```

Section 3.3

```
def proterm(i, value, x):
    pro = 1;
    for j in range(i):
        pro = pro * (value - x[j]);
    return pro;

# divided difference table
def dividedDiffTable(x, y, n):
    for i in range(1, n):
        for j in range(n - i):
            y[j][i] = ((y[j][i - 1] - y[j + 1][i - 1]) /
                                    (x[j] - x[i + j]));
    return y;

# Function for applying Newton's divided difference formula
```

```
def applyFormula(value, x, y, n):
    sum = y[0][0];
    for i in range(1, n):
        sum = sum + (proterm(i, value, x) * y[0][i]);
    return sum;

# Function for displaying divided difference table
def printDiffTable(y, n):
    for i in range(n):
        for j in range(n - i):
            print(y[i][j], "\t",
                              end = " ");
        print("");
  #partical codes refers to https://www.geeksforgeeks.org/newtons-divided-differenc
```

Question 1a

```
n = 4;
y = [[0 for i in range(10)]
        for j in range(10)];
x = [ 8.1, 8.3, 8.6, 8.7 ];

y[0][0] = 16.94410;
y[1][0] = 17.56492;
y[2][0] = 18.50515;
y[3][0] = 18.82091;

y=dividedDiffTable(x, y, n);
printDiffTable(y, n);
applyFormula(8.4, x, y, n)
```

```
16.9441          3.104099999999993       0.06000000000003336     -0.0020833333
17.56492         3.13410000000001        0.05875000000002649
18.50515         3.1576000000000204
18.82091
17.877142499999998
```

Question 1b

```
n = 4;
y = [[0 for i in range(10)]
        for j in range(10)];
x = [ 0.6, 0.7, 0.8, 1.0 ];

y[0][0] = -0.17694460;
y[1][0] = 0.01375227;
y[2][0] = 0.22363362;
y[3][0] = 0.65809197;

y=dividedDiffTable(x, y, n);
```

```
printDiffTable(y, n);
applyFormula(0.9, x, y, n)
```

```
-0.1769446      1.9069687000000006      0.9592239999999871      -1.7857412499
0.01375227      2.098813499999998       0.24492750000000726
0.22363362      2.1722917500000003
0.65809197
0.4419850024999998
```

Question 8a

```
n = 5;
y = [[0 for i in range(10)]
        for j in range(10)];
x = [ 0.0, 0.1, 0.3, 0.6, 1.0 ];

y[0][0] = -6.00000;
y[1][0] = -5.89483;
y[2][0] = -5.65014;
y[3][0] = -5.17788;
y[4][0] = -4.28172;

y=dividedDiffTable(x, y, n);
printDiffTable(y, n);
```

```
-6.0    1.051700000000002       0.5724999999999832      0.21500000000004166
-5.89483        1.223449999999997       0.7015000000000082      0.27801587301
-5.65014        1.5742000000000012      0.9517142857142844
-5.17788        2.2404
-4.28172
```

Question 8b

```
n = 6;
y = [[0 for i in range(10)]
        for j in range(10)];
x = [ 0.0, 0.1, 0.3, 0.6, 1.0, 1.1 ];

y[0][0] = -6.00000;
y[1][0] = -5.89483;
y[2][0] = -5.65014;
y[3][0] = -5.17788;
y[4][0] = -4.28172;
y[5][0] = -3.99583;

y=dividedDiffTable(x, y, n);
printDiffTable(y, n);
```

```
-6.0     1.051700000000002       0.5724999999999832      0.21500000000004166
-5.89483         1.223449999999997       0.7015000000000082      0.27801587301
-5.65014         1.5742000000000012      0.9517142857142844      0.35660714285
-5.17788         2.2404          1.2369999999999892
-4.28172         2.858899999999995
-3.99583
```