```
Alexander Winkles Computer Project #3
09/28/2016

-------------------------------------------------------------------------------
Here is a copy of my code:
-------------------------------------------------------------------------------

type newtondivdiff.m

function p = newtondivdiff(xx,yy,q1,q2,q3)
%p(x) = newtondivdiff(xx,yy,h)
%
%This is an algorithm developed by Alexander Winkles that takes data and
%utilizes divided differences to generate the Newton interpolating
%polynomial of the given data.
%
%The output p(x) may be used for evaluation the polynomial in various ways.
%
%xx : the nodes
%yy : the results
%q1 : prints divided difference table if q1 == 1
%q2 : prints the interpolated polynomial if q2 == 1
%q3 : prints figures to compare data to interpolating polynomial if q3 == 1

format rat

%Builds a matrix of the divided difference values, including f(x_i)
%values.

l = length(xx);

t = zeros(l,l);

t(:,1) = yy';

for j = 2 : l
    for i = 1 : (l - j + 1)
        t(i,j) = (t(i+1,j-1) - t(i,j-1))./(xx(i+j-1) - xx(i));
    end;
end;

%Uses values from the first row of the divided difference table to
%generate the Newton interpolating polynomial.

q =@(x) 0;
v =@(x) 1;

for j = 1 : l
    if j == 1;
        q =@(x) t(1,j);
    else
        u = t(1,j);
        v =@(x) v(x).*(x - xx(j-1));
        q =@(x) q(x) + u.*v(x);
```

```
    end;
end;

%Generates the same polynomial as above for printing purposes. (find out
%how to remove

b = 0;
a = 1;

for j = 1 : l
    if j == 1;
        b = t(1,j);
    else
        syms x;
        u = t(1,j);
        a = a*(x - xx(j-1));
        b = b + u*a;
    end;
end;

%Print results.

if q1 == 1
    fprintf('\nThe divided difference table is as follows:\n\n')
    disp(t)
end;

if q2 == 1
    fprintf('\nThe Newton interpolating polynomial is:\n\n %s\n\nUse the following to further evaluate t
end;

p =@(x) q(x);

%Plots points and the interpolating polynomial.

if q3 == 1
    figure('Name','Data points')
    scatter(xx,yy);

    figure('Name','Data points on polynomial')
    %sorted = sort(xx);
    %ss = sorted(2) : 0.01: sorted(length(sorted)-1);
    rr = min(xx) : 0.1: max(xx);
    %plot(ss,q(ss));
    plot(rr,q(rr));
    hold on
    plot(xx,yy,'+')
    hold off

    figure('Name','Interpolating polynomial over a large domain')
    domain = 0 : 1 : 500;
    range = q(domain);
    plot(domain,range)
    hold on
```

```
    plot(xx,yy,'+')
    hold off
end;

---------------------------------------------------------------------------
Problem 1:
---------------------------------------------------------------------------

Here are the data points given from the problem, with 0 being defined as 1958.
Likewise, we made y values integers, so 5 cents is 5.

xx = [5 10 13 16 20 23.25 23.75 27 30 33 37 41 43];

yy = [5 6 8 10 15 18 20 22 25 29 32 33 34];

Now we call the function to obtain a interpolating polynomial to work with.

z = newtondivdiff(xx,yy,0,0,1)

z =

    @(x)q(x)

To find when P(x) = 100 ($1), we define a new polynomial 'poly' that we may apply
    the bisection method to.

poly =@(x) z(x) - 100

poly =

    @(x)z(x)-100

bisec(poly,20,50,1e-5,100,1)

Step                  Result
---                 --------
0                  35.000000
1                  42.500000
2                  46.250000
3                  44.375000
4                  43.437500
5                  42.968750
6                  43.203125
7                  43.085938
8                  43.144531
9                  43.115234
10                  43.100586
11                  43.107910
12                  43.111572
13                  43.113403
14                  43.114319
15                  43.114777
16                  43.115005
17                  43.114891
```

```
18                  43.114834
19                  43.114805
20                  43.114820
21                  43.114827
22                  43.114830
23                  43.114828
24                  43.114828
25                  43.114828
26                  43.114828
27                  43.114828
28                  43.114828
29                  43.114828

The solution is 4.311483e+01. The computation was a success after 29 iterations!

poly(43.114828)

ans =

    -1.681689791439567e-04

Now we do the same for P(x) = 10000 ($100).

poly =@(x) z(x) - 10000

poly =

    @(x)z(x)-10000

bisec(poly,20,5000,1e-5,100,1)

Step                  Result
---                 --------
0                  2510.000000
1                  1265.000000
2                  642.500000
3                  331.250000
4                  175.625000
5                  97.812500
6                  58.906250
7                  39.453125
8                  49.179688
9                  44.316406
10                  46.748047
11                  45.532227
12                  44.924316
13                  45.228271
14                  45.380249
15                  45.304260
16                  45.342255
17                  45.361252
18                  45.351753
19                  45.356503
20                  45.354128
```

```
21                    45.355315
22                    45.354722
23                    45.355018
24                    45.354870
25                    45.354796
26                    45.354759
27                    45.354740
28                    45.354749
29                    45.354745
30                    45.354747
31                    45.354746
32                    45.354746
33                    45.354746
34                    45.354746
35                    45.354746
36                    45.354746
37                    45.354746

The solution is 4.535475e+01. The computation was a success after 37 iterations!

poly(45.354746)

ans =

     5.902188568143174e-04

Unfortunately, these results do not make sense with the given data. It is
thus concluded that Newton interpolation fails outside of the domain of given
points.

---------------------------------------------------------------------------
Problem 2:
---------------------------------------------------------------------------

Now we will test several functions with various n's to obtain their divided
    difference tables.

f =@(x) x.^2 + 2

f =

    @(x)x.^2+2

x = -4 : 1 : 4;
y = f(x);
newtondivdiff(x,y,1,0,0)

The divided difference table is as follows:

  Columns 1 through 8

      18              -7               1               0               0               0               0
      11              -5               1               0               0               0               0
       6              -3               1               0               0               0               0
```

```
       3            -1             1             0             0             0             0
       2             1             1             0             0             0             0
       3             3             1             0             0             0             0
       6             5             1             0             0             0             0
      11             7             0             0             0             0             0
      18             0             0             0             0             0             0

  Column 9

       0
       0
       0
       0
       0
       0
       0
       0
       0


ans =

    @(x)q(x)

g =@(x) x.^3 - 7

g =

    @(x)x.^3-7

x = -6 : 1 : 6;
y = g(x);
newtondivdiff(x,y,1,0,0)

The divided difference table is as follows:

  Columns 1 through 8

    -223            91           -15             1             0             0             0
    -132            61           -12             1             0             0             0
     -71            37            -9             1             0             0             0
     -34            19            -6             1             0             0             0
     -15             7            -3             1             0             0             0
      -8             1             0             1             0             0             0
      -7             1             3             1             0             0             0
      -6             7             6             1             0             0             0
       1            19             9             1             0             0             0
      20            37            12             1             0             0             0
      57            61            15             0             0             0             0
     118            91             0             0             0             0             0
     209             0             0             0             0             0             0

  Columns 9 through 13
```

```
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0
        0              0              0              0              0


ans =

    @(x)q(x)

x = -3 : 1 : 3;
y = g(x);
newtondivdiff(x,y,1,0,0)

The divided difference table is as follows:

     -34             19             -6              1              0              0              0
     -15              7             -3              1              0              0              0
      -8              1              0              1              0              0              0
      -7              1              3              1              0              0              0
      -6              7              6              0              0              0              0
       1             19              0              0              0              0              0
      20              0              0              0              0              0              0


ans =

    @(x)q(x)

h =@(x) sin(x)

h =

    @(x)sin(x)

x = -3 : 1 : 3;
y = h(x);
newtondivdiff(x,y,1,0,0)

The divided difference table is as follows:

    -441/3125      -729/949        887/2122        -60/5773      -151/5095       127/21426         0
    -401/441        347/5116       364/941        -409/3172         0            127/21426         0
   -1327/1577      1327/1577         0            -409/3172       151/5095          0              0
        0          1327/1577      -364/941         -60/5773         0               0              0
```

```
    1327/1577       347/5116      -887/2122          0              0              0              0
     401/441       -729/949           0              0              0              0              0
     441/3125          0              0              0              0              0              0


ans =

    @(x)q(x)

---------------------------------------------------------------------------
Problem 3:
---------------------------------------------------------------------------

We define the function as follows:

f =@(x) 1./(1+25*x.^2)

f =

    @(x)1./(1+25*x.^2)

First we test n = 5:

x = -1 : 0.5 : 1;
y = f(x);
newtondivdiff(x,y,1,0,1)

The divided difference table is as follows:

       1/26          75/377        575/377      -1250/377       1250/377
       4/29          50/29        -100/29        1250/377          0
       1            -50/29         575/377          0              0
       4/29         -75/377           0             0              0
       1/26            0              0             0              0


ans =

    @(x)q(x)

Now we test n = 10:

x = -1 : 0.25 : 1;
y = f(x);
newtondivdiff(x,y,0,0,1)

ans =

    @(x)q(x)

Finally, we test n = 15:

x = -1 : 0.1 : 1;
y = f(x);
```

```
newtondivdiff(x,y,0,0,1)

ans =

    @(x)q(x)

Now we produce a graph of the function for comparison. All graphs may be found
    in order after this log.

domain = [-1 1];

fplot(f,domain)
diary off
```

Figure 1: Polynomial interpolation when n = 5

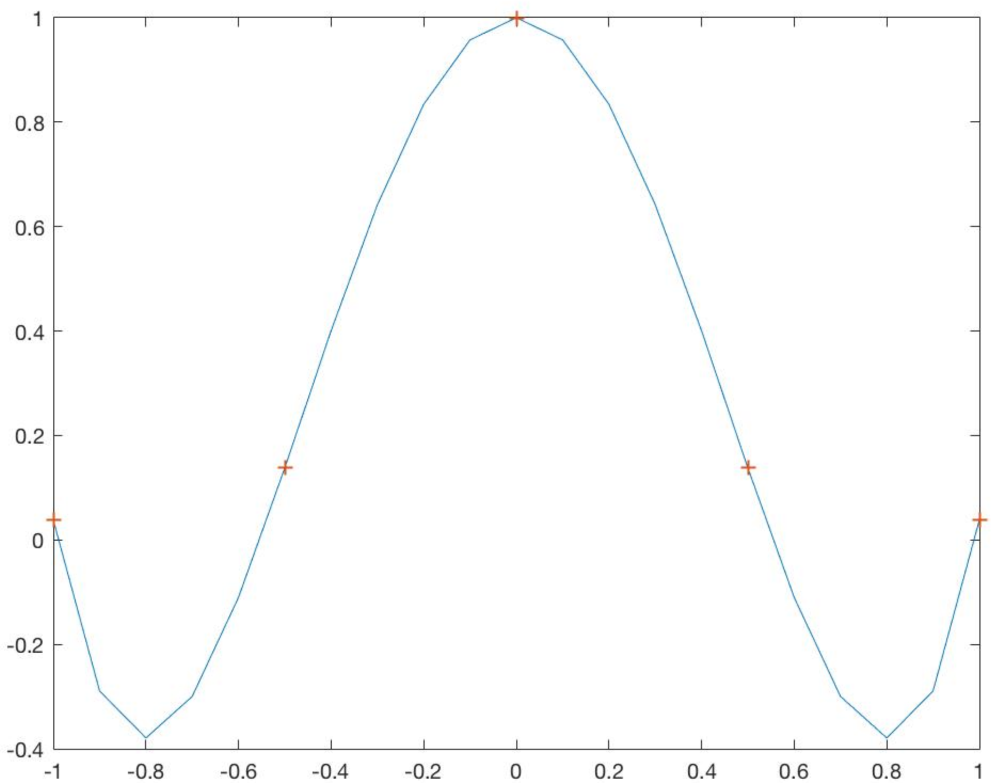

Figure 2: Polynomial interpolation when n = 10

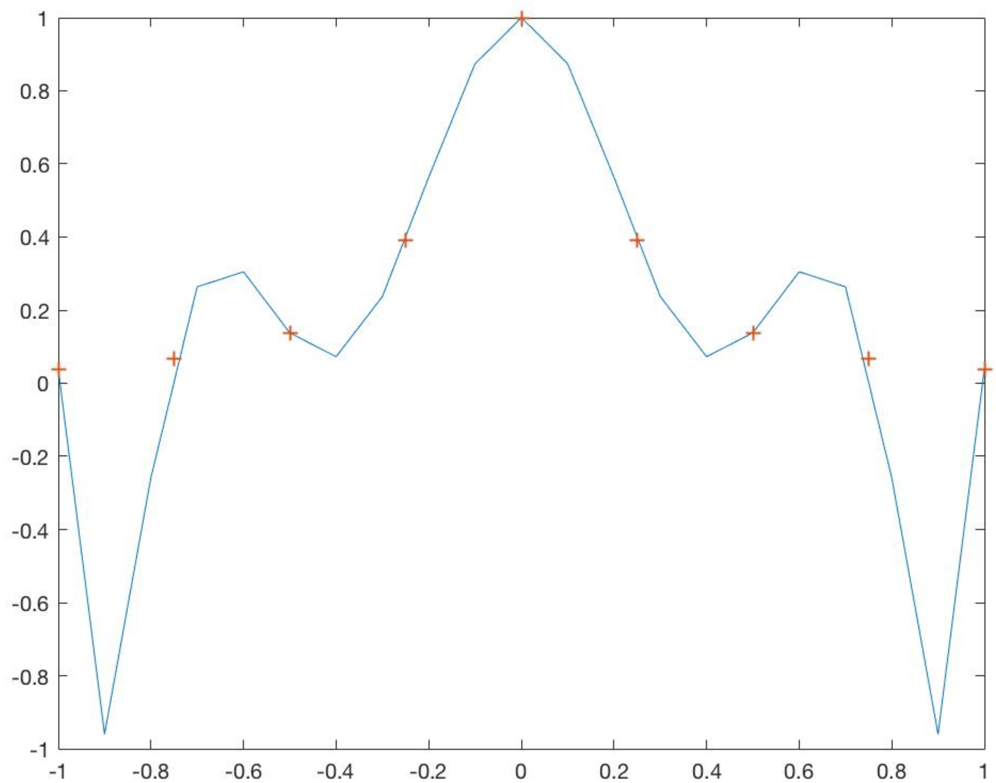

Figure 3: Polynomial interpolation when n = 15

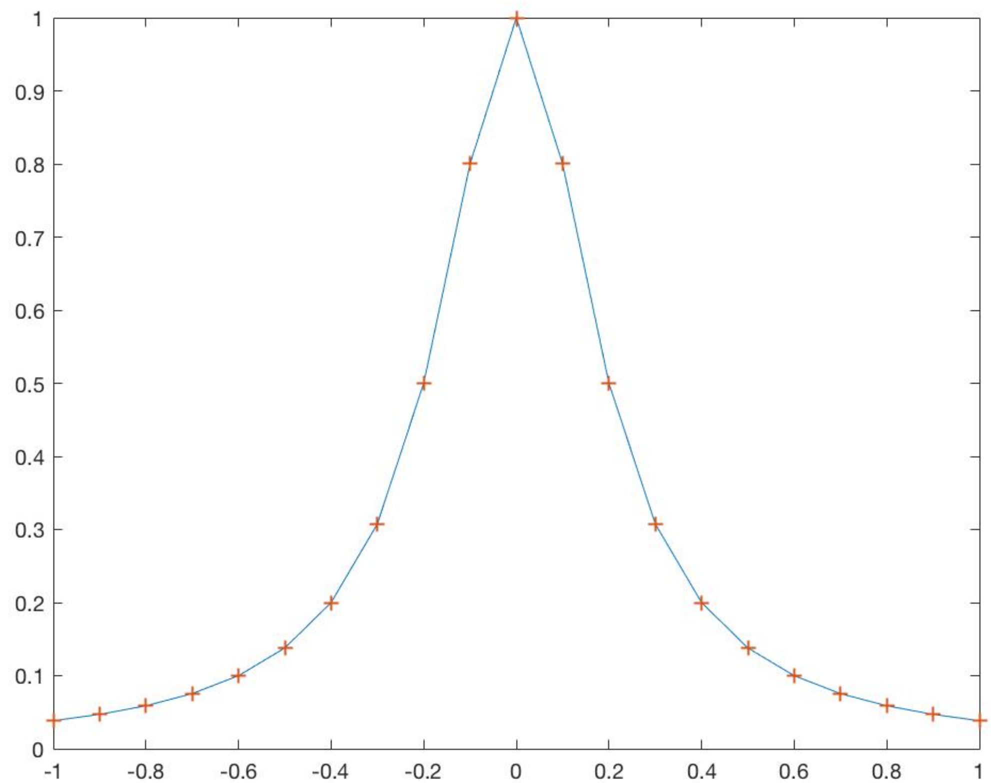

Figure 4: The graph of f(x)