Table of Contents

```
clear all;clc
```

Given Data

```
x = [1 2 3 5 7 8];
y = [3 6 19 99 291 444];
```

Newton's interpolation polynomials of

```
%%order 1
x1 = [3 5];
y1 = [19 99];

% x1 = [2 3];
% y1 = [6 19]; %I tried but not getting acurate results

d1 = divdiff(x1,y1);
pval1 = evnewt(d1,x1,4);
fprintf('\nf(x=4) using newtons interpolation polynomials of order 1
 is %f\n',pval1);
```

order 2

```
x2 = [ 2 3 5 ];
y2 = [ 6 19 99 ];     % I get the value as 50 the estimated correct ans
 is 48 so close to it is 50

% x2 = [3 5 7];
% y2 = [19 99 291];     % I got the value as 45

d2 = divdiff(x2,y2);
pval2 = evnewt(d2,x2,4);
fprintf('\nf(x=4) using newtons interpolation polynomials of order 2
 is %f\n',pval2);
```

```
f(x=4) using newtons interpolation polynomials of order 2 is 50.000000
```

order 3

```
x3 = [ 2 3 5 7];
y3 = [6 19 99 291];
d3 = divdiff(x3,y3);
pval3 = evnewt(d3,x3,4);
fprintf('\nf(x=4) using newtons interpolation polynomials of order 3
 is %f\n',pval3);
```

```
f(x=4) using newtons interpolation polynomials of order 3 is 48.000000
```

order 4

```
x4 = [ 2 3 5 7 8];
y4 = [6 19 99 291 444];
d4 = divdiff(x4,y4);
pval4 = evnewt(d4,x4,4);
fprintf('\nf(x=4) using newtons interpolation polynomials of order 4
 is %f\n',pval4);
```

```
f(x=4) using newtons interpolation polynomials of order 4 is 48.000000
```

all points

```
x5 = [ 1 2 3 5 7 8];
y5 = [3 6 19 99 291 444];
d5 = divdiff(x5,y5);
pval5 = evnewt(d5,x5,4);
fprintf('\nf(x=4) using newtons interpolation polynomials using all is
 %f\n',pval5);
```

divided difference function which returns values of d

```
function d = divdiff(x,y)
% d = divdiff(x,y)
% compute coefficients of Newton form of interpolating polynomial
% x: vector of nodes
% y: vector of y-values at nodes
% d: vector of Newton coefficients
%    d = [ f[x_1,...,x_n], f[x2,...,x_n], ..., f[x_n] ]
% use evnewt to evaluate interpolating polynomial

n = length(x);
d = y;
for k=1:n-1
  for i=1:n-k
    d(i) = (d(i+1)-d(i))/(x(i+k)-x(i));
  end
end
end


%%Function which finds the value of given point using the d from
 divided
%%difference
function p = evnewt(d,x,t)
% y = evnewt(d,x,t)
% evaluate Newton form of interpolating polynomial
% d: vector of divided difference coefficients as computed by divdiff
% x: vector of nodes
% t: vector of evaluation points
% p: vector of values of interpolating polynomial
%


p = d(1)*ones(size(t));
for i=2:length(d)
  p = p.*(t-x(i)) + d(i);
end
end
```

```
f(x=4) using newtons interpolation polynomials of order 1 is 59.000000
```

Published with MATLAB® R2018b