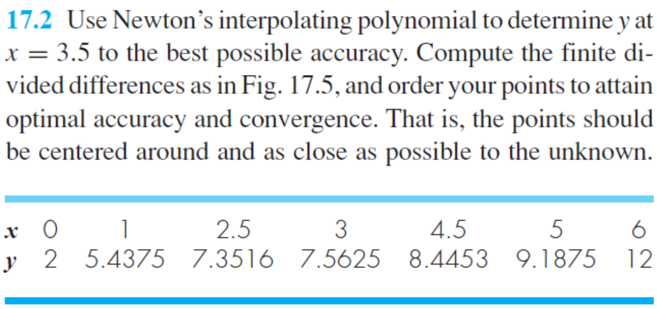
A. 每一題，請附上原始碼和執行畫面，以及必要之解說。

B. 你的 Word 檔應貼上程式碼的截圖或 M file 內容（非在 word 中重新 keyin 程式碼）。另外，再將各題 M files 與 Word 檔壓縮成 ZIP 後，再上傳。 所有檔名請用學號命名。

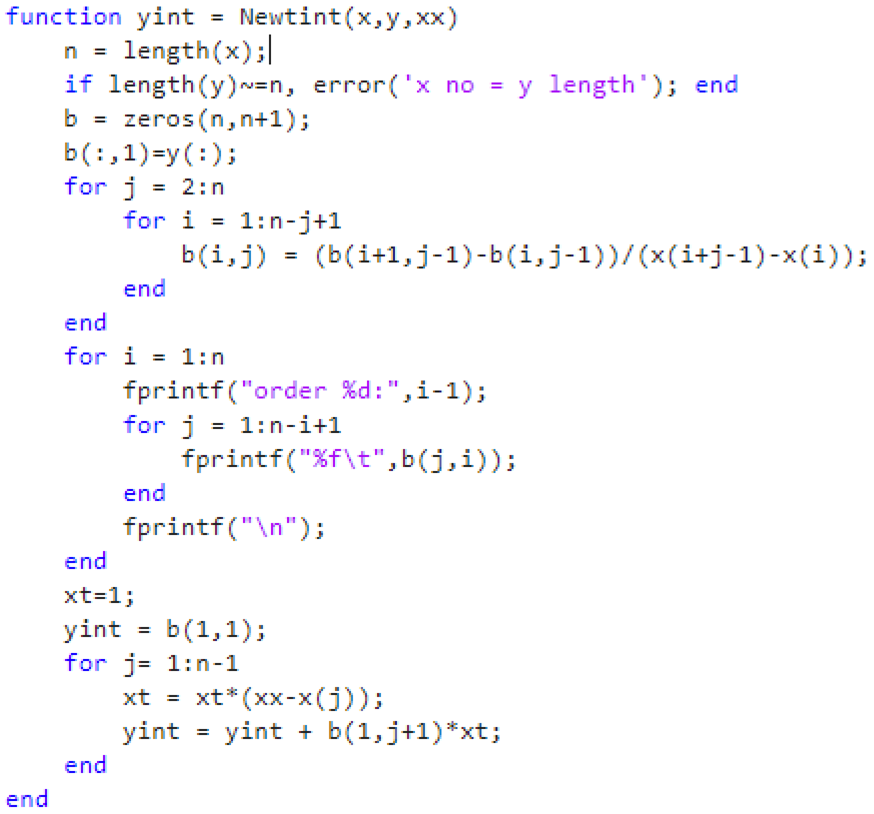
C. 每一題輸入值皆應寫入 M file。



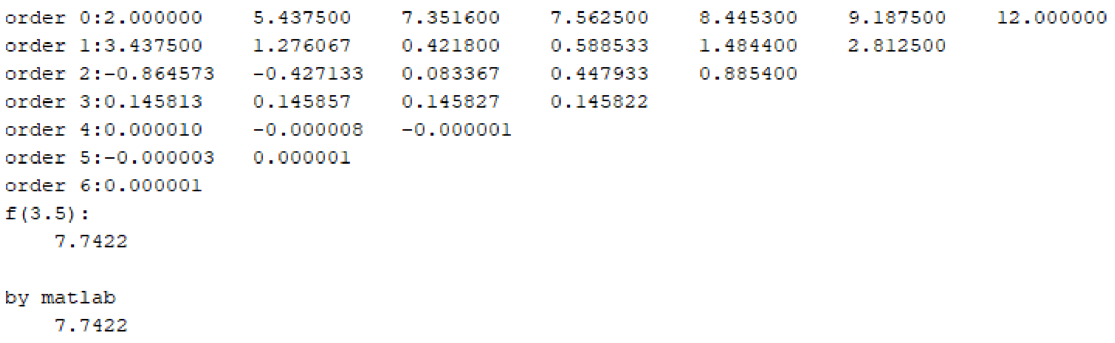
17.2 Use Newton's interpolating polynomial to determine y at $x = 3.5$ to the best possible accuracy. Compute the finite divided differences as in Fig. 17.5, and order your points to attain optimal accuracy and convergence. That is, the points should be centered around and as close as possible to the unknown.

x	0	1	2.5	3	4.5	5	6
y	2	5.4375	7.3516	7.5625	8.4453	9.1875	12

1.

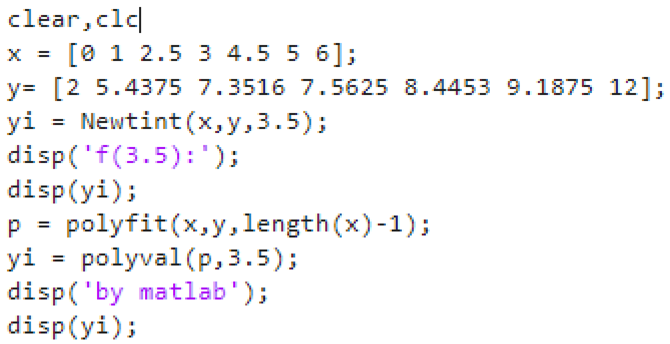


```
function yint = Newtint(x,y,xx)
    n = length(x);
    if length(y)~=n, error('x no = y length'); end
    b = zeros(n,n+1);
    b(:,1)=y(:);
    for j = 2:n
        for i = 1:n-j+1
            b(i,j) = (b(i+1,j-1)-b(i,j-1))/(x(i+j-1)-x(i));
        end
    end
    for i = 1:n
        fprintf("order %d:",i-1);
        for j = 1:n-i+1
            fprintf("%f\t",b(j,i));
        end
        fprintf("\n");
    end
    xt=1;
    yint = b(1,1);
    for j= 1:n-1
        xt = xt*(xx-x(j));
        yint = yint + b(1,j+1)*xt;
    end
end
```



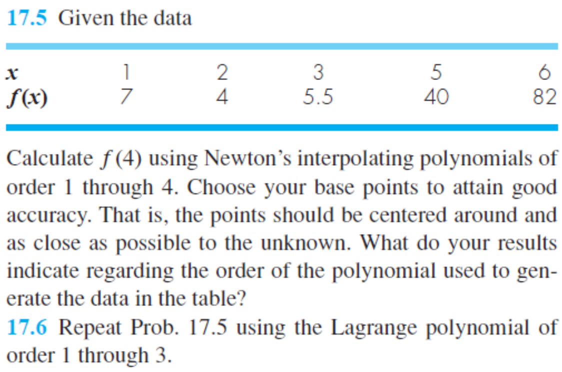
```
order 0:2.000000    5.437500    7.351600    7.562500    8.445300    9.187500    12.000000
order 1:3.437500    1.276067    0.421800    0.588533    1.484400    2.812500
order 2:-0.864573   -0.427133   0.083367    0.447933    0.885400
order 3:0.145813    0.145857    0.145827    0.145822
order 4:0.000010    -0.000008   -0.000001
order 5:-0.000003   0.000001
order 6:0.000001
f(3.5):
    7.7422

by matlab
    7.7422
```



```
clear,clc
x = [0 1 2.5 3 4.5 5 6];
y= [2 5.4375 7.3516 7.5625 8.4453 9.1875 12];
yi = Newtint(x,y,3.5);
disp('f(3.5):');
disp(yi);
p = polyfit(x,y,length(x)-1);
yi = polyval(p,3.5);
disp('by matlab');
disp(yi);
```

以牛頓多項式差分法把 x=3.5 時，y 值推測出，結果為 7.7422。牛頓多項式差分法主要把 n 筆資料換成 n-1 個 order，每個 order 由前一 order 的兩項以斜率的觀念依序計算，題目雖然暗示資料可能要排序，不過其實嘗試後答案都會是一樣的，最後則是用 matlab 的 polyfit 和 polyval 驗算。



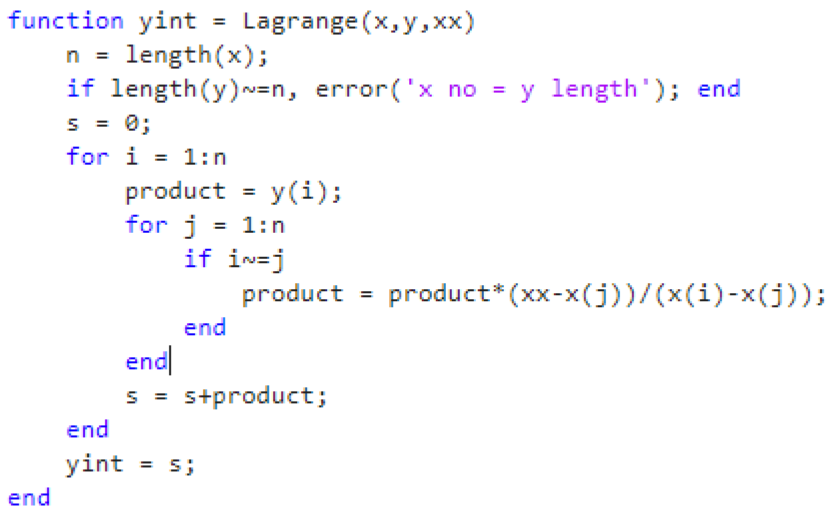
17.5 Given the data

x	1	2	3	5	6
$f(x)$	7	4	5.5	40	82

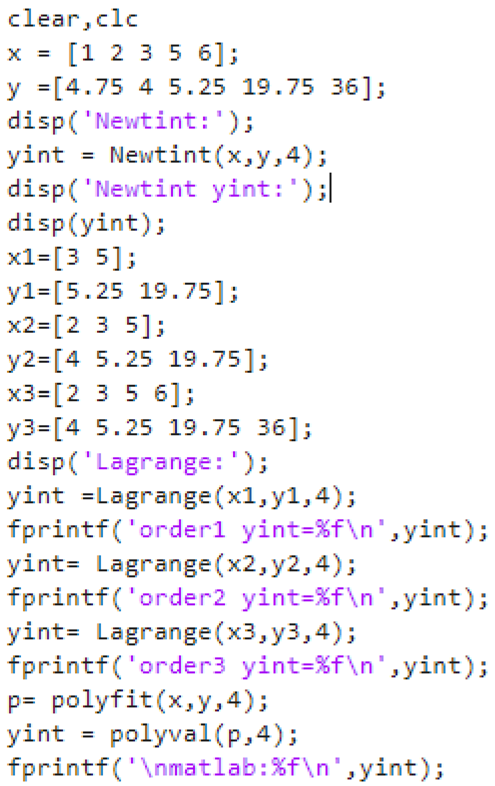
Calculate $f(4)$ using Newton's interpolating polynomials of order 1 through 4. Choose your base points to attain good accuracy. That is, the points should be centered around and as close as possible to the unknown. What do your results indicate regarding the order of the polynomial used to generate the data in the table?

17.6 Repeat Prob. 17.5 using the Lagrange polynomial of order 1 through 3.

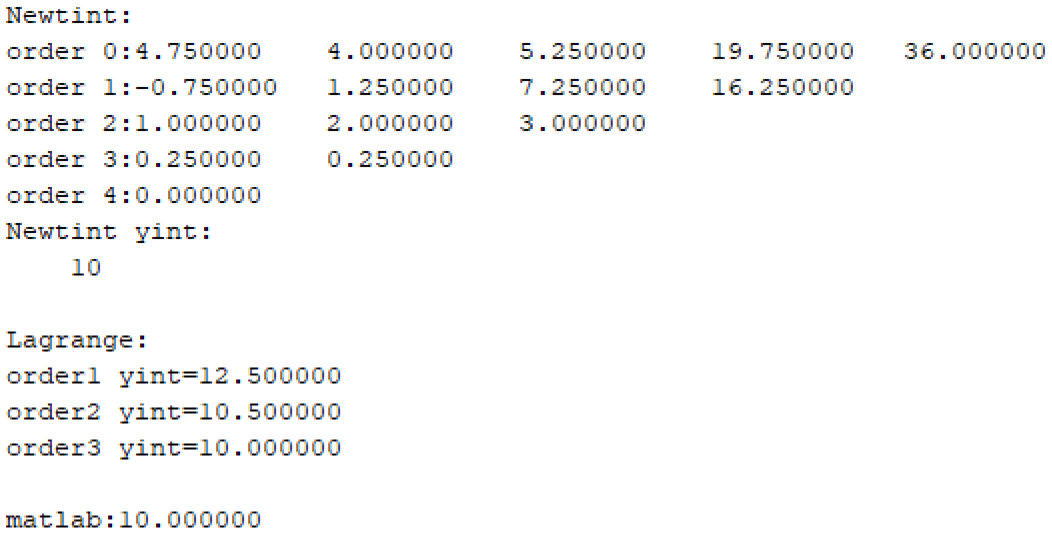
2.



```
function yint = Lagrange(x,y,xx)
    n = length(x);
    if length(y)~=n, error('x no = y length'); end
    s = 0;
    for i = 1:n
        product = y(i);
        for j = 1:n
            if i~=j
                product = product*(xx-x(j))/(x(i)-x(j));
            end
        end
        s = s+product;
    end
    yint = s;
end
```



```
clear,clc
x = [1 2 3 5 6];
y =[4.75 4 5.25 19.75 36];
disp('Newtint:');
yint = Newtint(x,y,4);
disp('Newtint yint:');
disp(yint);
x1=[3 5];
y1=[5.25 19.75];
x2=[2 3 5];
y2=[4 5.25 19.75];
x3=[2 3 5 6];
y3=[4 5.25 19.75 36];
disp('Lagrange:');
yint =Lagrange(x1,y1,4);
fprintf('order1 yint=%f\n',yint);
yint= Lagrange(x2,y2,4);
fprintf('order2 yint=%f\n',yint);
yint= Lagrange(x3,y3,4);
fprintf('order3 yint=%f\n',yint);
p= polyfit(x,y,4);
yint = polyval(p,4);
fprintf('\nmatlab:%f\n',yint);
```



```
Newtint:
order 0:4.750000    4.000000    5.250000    19.750000   36.000000
order 1:-0.750000   1.250000    7.250000    16.250000
order 2:1.000000    2.000000    3.000000
order 3:0.250000    0.250000
order 4:0.000000
Newtint yint:
    10

Lagrange:
order1 yint=12.500000
order2 yint=10.500000
order3 yint=10.000000

matlab:10.000000
```

Newtint 的函式是沿用第一題，一樣經過 matlab 驗算 f(4)的結果算出來是 10。多了 lagrange 的函式，概念是透過 x 和資料產生加權係數 L，最後和牛頓一樣 n 筆資料會產生 n-1 階多項式。