Given $m\frac{d^2}{dt^2}x + c\frac{dx}{dt} + kx = 0$

with $x(0) = 1$ m and $\left.\frac{dx}{dt}\right|_{x=0} = 0$ m/s

---

Characteristic eqn

$m\lambda^2 + c\lambda + k = 0 \Rightarrow \boxed{\lambda = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m}}$

Case 1 $m = 2.5$ kg $c = 10.0$ kg/s $k = 10$ kg/s^2

$\therefore \lambda = \frac{-10 \pm \sqrt{100 - 4(2.5)10}}{5}$

critical damping $c^2 = 4mk$

$\boxed{\lambda_{1,2} = -2}$ repeated roots

$\therefore \boxed{x(t) = A_1 e^{-2t} + A_2 t e^{-2t}}$

Now $x(0) = 1 = A_1 + 0 \quad \therefore \boxed{A_1 = 1}$

$\frac{d}{dt}x = -2A_1 e^{-2t} - 2A_2 t e^{-2t} + A_2 e^{-2t}$

$= A_2(1 - 2t)e^{-2t} - 2A_1 e^{-2t}$

and $\left.\frac{d}{dt}x\right|_{t=0} = A_2 - 2A_1 = 0 \quad \therefore \boxed{A_2 = 2A_1 = 2}$

$\therefore \boxed{x(t) = (1 + 2t)e^{-2t}}$ ans

Case 2 $m = 2.5$ kg $c = 12.5$ kg/s $k = 10$ kg/s^2

$\therefore \lambda = \frac{-12.5 \pm \sqrt{156.25 - 100}}{5}$

over damped $c^2 > 4mk$

$= \frac{-12.5 \pm 7.5}{5} \quad \therefore \boxed{\begin{array}{l}\lambda_1 = -1 \\ \lambda_2 = -4\end{array}}$

$\therefore \boxed{x(t) = A_1 e^{-t} + A_2 e^{-4t}}$

Now $x(0) = 1 = A_1 + A_2 \qquad A_1 = 1 - A_2$

$\frac{d}{dt}x = -A_1 e^{-t} - 4A_2 e^{-4t}$

$-4A_2 = 1 - A_2$

$\boxed{\begin{array}{l}A_2 = -\frac{1}{3} \\ A_1 = \frac{4}{3}\end{array}}$

and $\left.\frac{d}{dt}x\right|_{t=0} = -A_1 - 4A_2 = 0 \qquad A_1 = -4A_2$

$\therefore \boxed{x(t) = \frac{1}{3}\left(4e^{-t} - e^{-4t}\right)}$ ans

Case 3 $m = 2.5\,kg$ $c = 6\,kg/s$ $k = 10\,kg/s^2$

under damped $c^2 < 4mk$

$$\lambda = \frac{-6 \pm \sqrt{36-100}}{5}$$

$$= \frac{-6 \pm 8i}{5} \quad \therefore \quad \lambda_1 = -1.2 + 1.6i, \quad \lambda_2 = -1.2 - 1.6i$$

$$\therefore \boxed{x(t) = A_1 e^{-1.2t} \sin 1.6t + A_2 e^{-1.2t} \cos 1.6t}$$

now $x(0) = 1 = 0 + A_2$ $\boxed{A_2 = 1}$

$$\frac{d}{dt}x = 1.6A_1 e^{-1.2t} \cos 1.6t - 1.2A_1 e^{-1.2t} \sin 1.6t$$
$$- 1.6A_2 e^{-1.2t} \sin 1.6t - 1.2A_2 e^{-1.2t} \cos 1.6t$$

$$\left.\frac{d}{dt}x\right|_{t=0} = 1.6A_1 - 0 - 0 - 1.2A_2 = 0$$

$$1.6A_1 = 1.2A_2 = 1.2$$

$$\boxed{A_1 = \frac{1.2}{1.6} = \frac{3}{4}}$$

$$\boxed{x(t) = e^{-1.2t}\left[\frac{3}{4}\sin 1.6t + \cos 1.6t\right]}$$ ans

Case 4 $m = 2.5\,kg$ $c = 0\,kg/s$ $k = 10\,kg/s^2$

undamped $c^2 = 0$

$$\lambda = \frac{-0 \pm \sqrt{0-100}}{5}$$

$$\boxed{\lambda = \pm 2i}$$

$$\therefore \boxed{x(t) = A_1 \sin 2t + A_2 \cos 2t}$$

now $x(0) = 1 = 0 + A_2$ $\boxed{A_2 = 1}$

$$\frac{d}{dt}x = 2A_1 \cos 2t - 2A_2 \sin 2t$$

$$\left.\frac{d}{dt}x\right|_{x=0} = 2A_1 - 0 = 0 \quad \therefore \boxed{A_1 = 0}$$

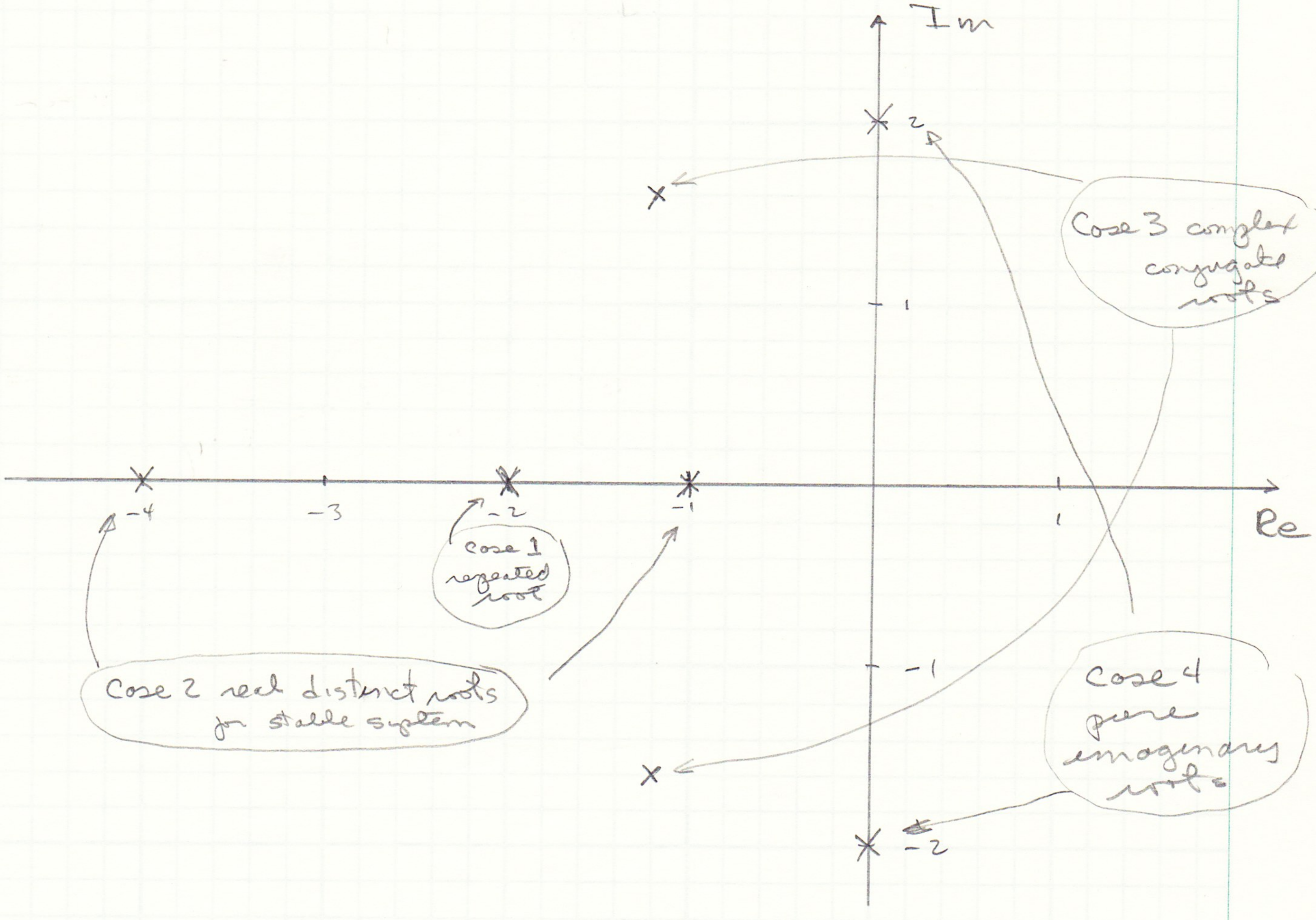
$$\therefore \boxed{x(t) = \cos 2t}$$ ans

→ see matlab plots for these functions mech_sys1_cs3 (↑ case #3)

Note: The responses observed here are typical of any 2nd order LTI system

1. Critical Damping — repeated roots
2. Overdamped Case - real distinct roots
3. Underdamped Case - complex conjugate roots
4. Undamped Case - pure imaginary roots

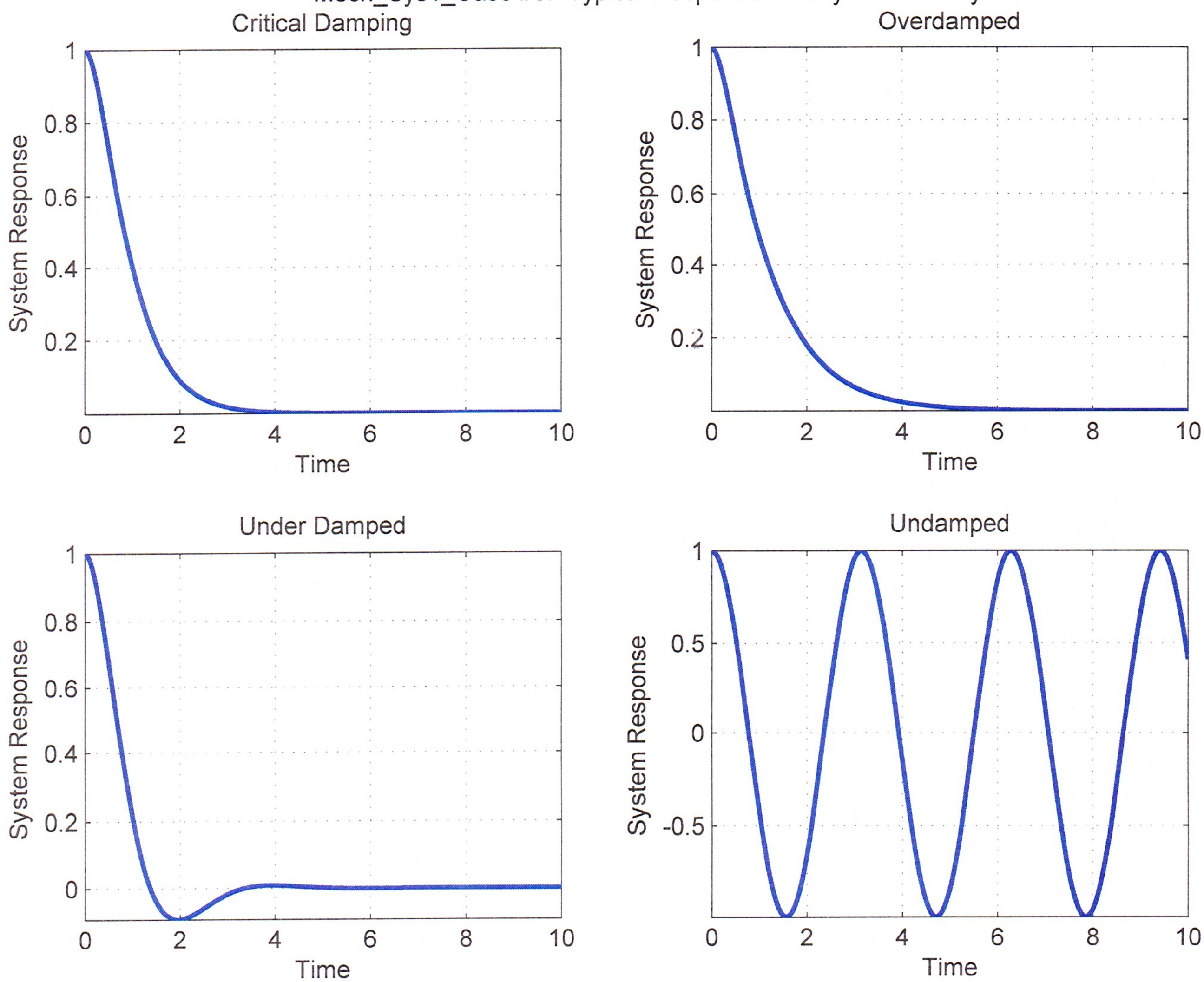
The time domain behavior observed in the Matlab plots can be correlated with the location of the roots in the complex plane.



→ Knowing the root location gives us a good idea of what the time domain response looks like.

→ For higher order systems, there is often a dominant root or set of complex conjugate roots that, after some transient time, gives a good approximation to the overall system behavior...

Mech_Sys1_Case #3: Typical Response for any 2nd order system



```
%
%   MECH_SYS1_CS3.M    Plot behavior of simple mechanical system (Case #3)
%                          (Typical response of any 2nd order system)
%
%   This file looks at the response of a simple mass-spring-dashpot mechanical
%   system with four different sets of values for the m, c, and k coefficients.
%   The coeficients were chosen to give the four different cases that can be observed
%   in any 2nd order system:
%     1. Critical Damping -- gives repeated roots of the characteristic polynomial
%     2. Overdamped Case -- real distinct roots (exponential decay for stable system)
%     3. Underdamped Case -- complex conjugate roots (damped sinusoid behavior)
%     4. Undamped Case -- no damping in system leads to pure oscillatory behavior
%
%   Note that the unforced 2nd order IVPs were solved analytically, and the goal here
%   is simply to plot the resultant responses to visualize the typical behavior that
%   can be expected for systems of this type.
%
%   File prepared by J. R. White, UMass-Lowell (Jan. 2014)
%

      clear all,   close all,   nfig = 0;
%
%   evaluate time domain solution for four systems (Case #2 data)
      t = linspace(0,10,201);
      x1 = (1 + 2*t).*exp(-2*t);
      x2 = (1/3)*(4*exp(-t) - exp(-4*t));
      x3 = exp(-1.2*t).*(cos(1.6*t) + (3/4)*sin(1.6*t));
      x4 = cos(2*t);
%
%   plot the solutions on a single page using the subplot command
      nfig = nfig+1;   figure(nfig)
      subplot(2,2,1),plot(t,x1,'LineWidth',2),title('Critical Damping'),grid
      xlabel('Time'),ylabel('System Response'),axis tight
      subplot(2,2,2),plot(t,x2,'LineWidth',2),title('Overdamped'),grid
      xlabel('Time'),ylabel('System Response'),axis tight
      subplot(2,2,3),plot(t,x3,'LineWidth',2),title('Under Damped'),grid
      xlabel('Time'),ylabel('System Response'),axis tight
      subplot(2,2,4),plot(t,x4,'LineWidth',2),title('Undamped'),grid
      xlabel('Time'),ylabel('System Response'),axis tight
%
%   put global title (edit placement once plot is generated)
      gtext('Mech\_Sys1\_Case #3:  Typical Response for any 2^{nd} order system')
%
%   end of simulation
```

Solve $y(k+2) - 3y(k+1) + 2y(k) = 3$

with $y(1) = 0$ and $y(2) = 1$ for $k = 1, 2, 3, \ldots$

- - - - -

general soln $y(k) = y_H(k) + y_p(k)$

(homogeneous) + (particular)

homo soln

assume $y_H \sim \lambda^k$

giving $(\lambda^2 - 3\lambda + 2)\lambda^k = 0$ $\Rightarrow$ $\lambda_1 = 1$, $\lambda_2 = 2$

$(\lambda - 2)(\lambda - 1) = 0$

$\therefore$ $y_H(k) = c_1 \lambda_1^k + c_2 \lambda_2^k$

$$y_H(k) = c_1 + c_2 2^k$$

part. soln . note — one usually assumes the form for y_p as the same as the forcing function
— however, since the homo soln contains a constant term, let's try

$y_p(k) = Ck$ (some constant times k)

upon subst, we have

$$C(k+2) - 3C(k+1) + 2Ck = 3$$

collecting terms $C(k - 3k + 2k) + C(2 - 3 + 0) = 3$

($\to 0$ for any C) $\therefore$ $C = -3$

$$\therefore \quad y(k) = c_1 + c_2 2^k - 3k$$

Unique soln

$y(1) = 0 = c_1 + c_2(2) - 3$

$c_1 + 2c_2 = 3$ (1)

$c_1 = 3 - 2c_2$

$c_1 = -1$

$y(2) = 1 = c_1 + c_2(4) - 6$

$c_1 + 4c_2 = 7$ (2)

Then (2)−(1) $\Rightarrow$ $2c_2 = 4$ or $c_2 = 2$

$\therefore$ final soln

$y(k) = -1 + 2(2^k) - 3k$

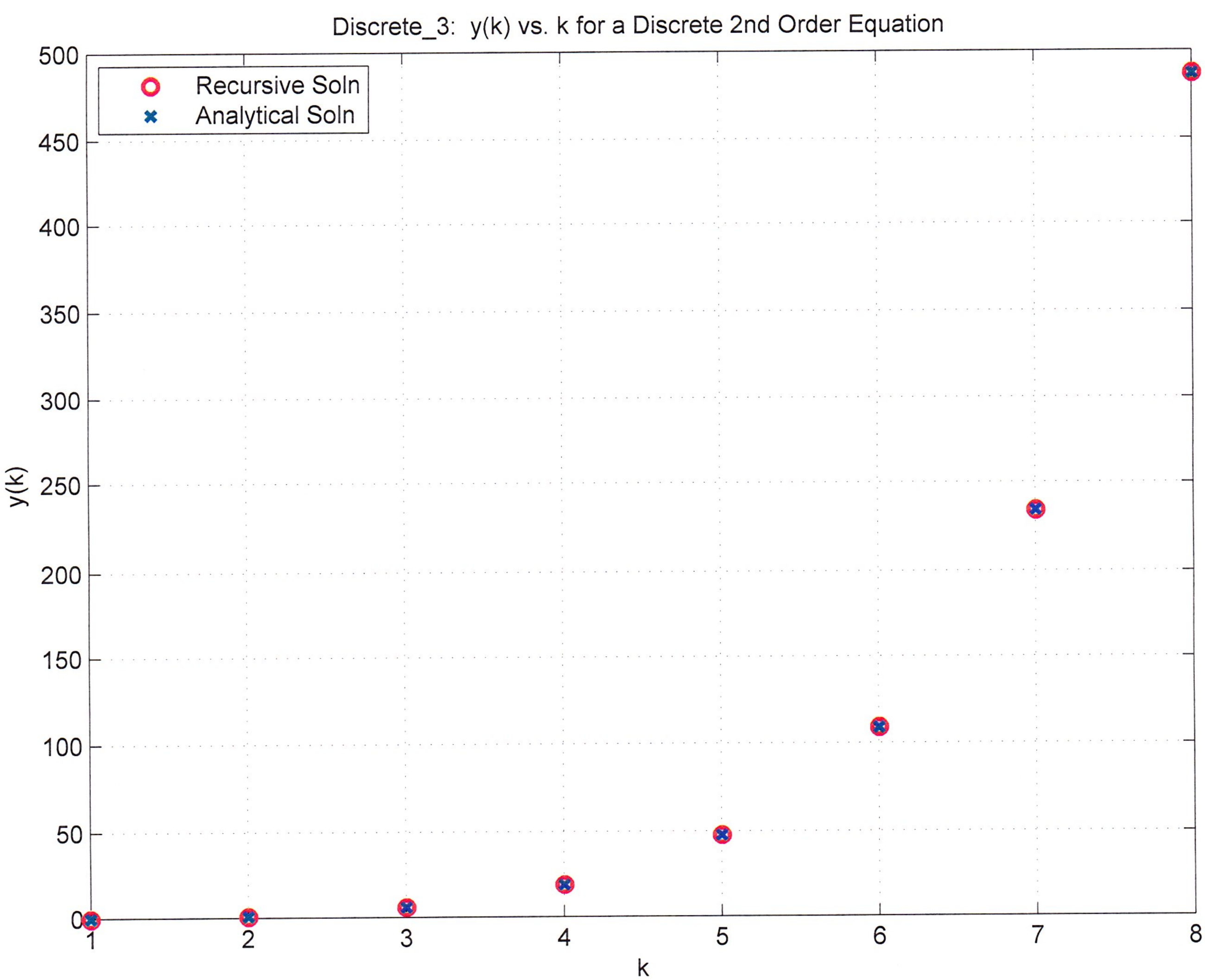
$$y(k) = 2^{k+1} - 3k - 1$$ ans

k	y(k)
1	0
2	1
3	6
4	19
⋮	

OK

```
%
%   DISCRETE_3.M    Plot discrete solution to 2nd Order Difference Eqn. (Version #3)
%
%   This file simply plots the solution to a specific 2nd order difference equation.
%   The analytical solution is plotted and this is compared to the solution obtained
%   by simply evaluating the recursive form of the original equation.  Both
%   techniques should give the same answer  -- let's see...
%
%   File written by J. R. White, UMass-Lowell  (Jan. 2014)
%

     clear all,   close all,   nfig = 0;
%
%   set initial conditions
     y1 = 0;    y2 = 1;
%
%   Solution Method #A   --  Recursive evaluation of discrete difference equation
     Nk = 8;  % this process has geometric growth so keep k relatively small
     ydA = zeros(1,Nk);
     ydA(1) = y1;   ydA(2) = y2;
     for k = 1:Nk-2
       ydA(k+2) = 3 + 3*ydA(k+1) - 2*ydA(k);
     end
%
%   Solution Method #B   --  Evaluation of discrete analytical solution
%    (note that this requires "dot arithmetic" because k is a vector)
     k = 1:Nk;
     ydB = 2.^(k+1) - 3*k - 1;
%
%   tabulate results (this illustrates the use of the fprintf command)
     disp(' ')
     fprintf('*** Discrete_3 -- Solution to Discrete Eqn (Two Methods) *** \n\n')
     fprintf('     k         ydA(k)     ydB(k) \n')
     for kk = 1:Nk
       fprintf('  %5i     %6i     %6i  \n',kk, ydA(kk), ydB(kk));
     end
%
%   plot the solutions
     nfig = nfig+1;   figure(nfig)
     plot(k,ydA,'ro',k,ydB,'bx','LineWidth',2),grid
     xlabel('k'),ylabel('y(k)')
     title('Discrete\_3:  y(k) vs. k for a Discrete 2nd Order Equation')
     xlabel('k'),ylabel('y(k)')
     legend('Recursive Soln','Analytical Soln','Location','NorthWest')
%
%   end of program
```


Discrete_3: y(k) vs. k for a Discrete 2nd Order Equation
Recursive Soln
Analytical Soln
y(k)
k
0
50
100
150
200
250
300
350
400
450
500
1
2
3
4
5
6
7
8

```
>> discrete_3

*** Discrete_3 -- Solution to Discrete Eqn (Two Methods) ***

      k        ydA(k)     ydB(k)
      1           0          0
      2           1          1
      3           6          6
      4          19         19
      5          48         48
      6         109        109
      7         234        234
      8         487        487
>>
```