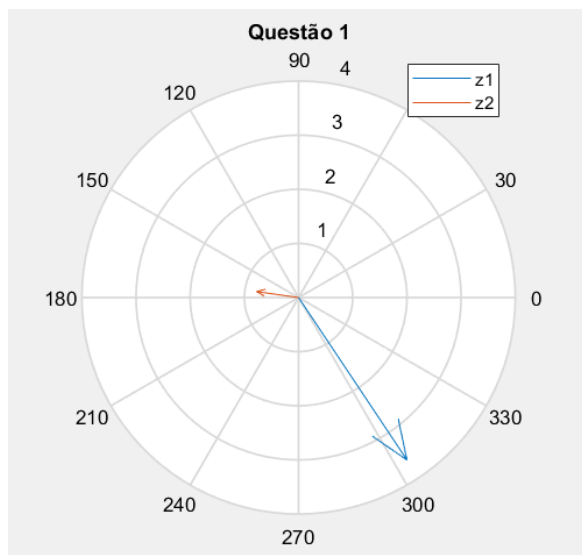


Questão 1

```

1 % Luiz Henrique Gariglio dos Santos – 2022421137
2
3 clear
4 close all
5 clc
6
7 %% 1
8
9 disp('Questão 1:');
10 z1 = 2-1j*3;
11
12 r = 3;
13 theta = pi/4;
14 [a,b] = pol2cart(r,theta);
15
16 z2 = a+1j*b;
17
18 figure;
19 compass(z1);
20 hold on;
21 compass(z2);
22 title('Questão 1');
23 legend('z1','z2');
24
25 z_mult = z1*z2;
26 fprintf('z1*z2 = %.3f + %.3fi\n', real(z_mult), imag(z_mult));
27 z_div = z1/z2;
28 fprintf('z1/z2 = %.3f + %.3fi\n', real(z_div), imag(z_div));

```



Command Window

```

Questão 1:
z1*z2 = -1.223 + 2.554i
z1/z2 = -3.060 + 3.422i

```

Figura 1: Resultados da Questão 1

Questão 2

```
1 %% 2
2 omega = 2 * pi * 3; % frequencia = 3;
3 alpha = log(0.5)/2; % decaimento de 50% a cada 2s
4 t = linspace(0, 5, 1000);
5 y = exp(alpha*t).* sin(omega*t);
6
7 figure;
8 plot(t,y);
9 title('Questão 2');
10 xlabel('Tempo[s]');
11 ylabel('Amplitude');
```

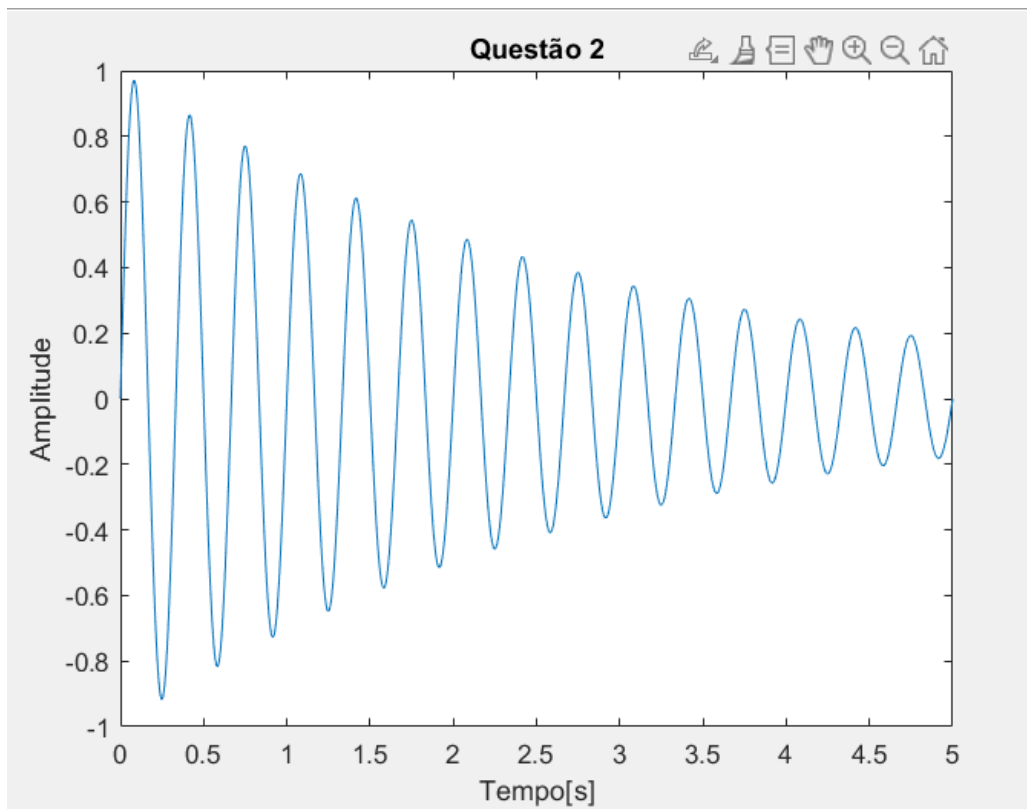


Figura 2: Resultado da Questão 2

Questão 3

```
1 %% 3
2 t = linspace(0, 3, 1000);
3 x1 = real(2*exp((-1+1j*2*pi)*t));
4 x2 = imag(3-exp((1-1j*2*pi)*t));
5 x3 = 3-imag(exp((1-1j*2*pi)*t));
6 figure
7 plot(t,x1,'b');
8 hold on;
9 plot(t,x2,'r');
10 plot(t,x3,'g');
11 legend('x1','x2','x3');
12 title('Questão 3');
```

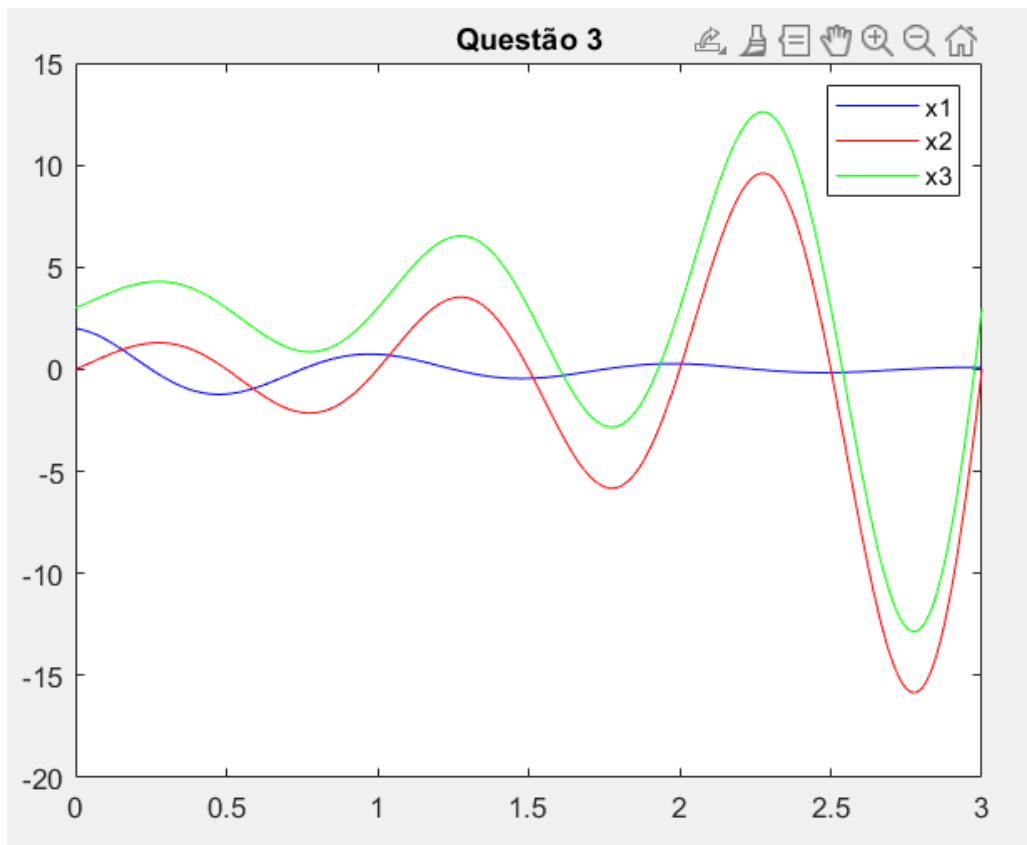


Figura 3: Resultado da Questão 3

Questão 4

```
1 %% 4
2 t = linspace(0, 5, 1000);
3 y = cos(t).*sin(20*t);
4 figure;
5 plot(t,y);
6 title('Questão 4');
7 xlabel('Tempo[s]');
8 ylabel('Amplitude');
```

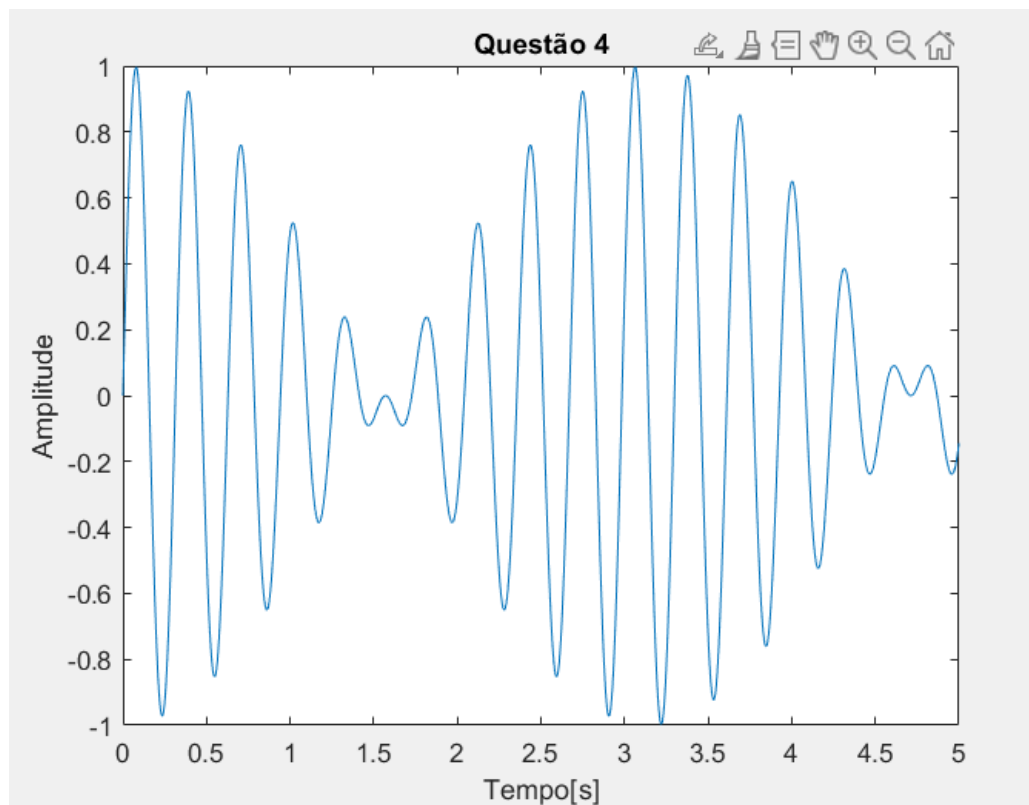


Figura 4: Resultado da Questão 4

Questão 5

```
1 %% 5
2
3 N = 6*[1 1];
4 D = conv([1 0], conv([1 1.46], [1 0.13]));
5 [R1,P1,K1] = residue(N,D);
6
7
8 disp('Questão 5.a:');
9 for i = 1:length(R1)
10     fprintf('%0.4f/(s - (%0.4f)) + ', R1(i), P1(i));
11 end
12 if ~isempty(K1)
13     for i = 1:length(K1)
14         fprintf('%0.4f*s^%d + ', K1(i), length(K1)-i);
15     end
16 end
17 fprintf('\b\b \n'); % Remove o último ' + '
18
19 N = [1 2 3];
20 D = [1 2 1];
21 [R2,P2,K2] = residue(N,D);
22
23 disp('Questão 5.b:');
24 for i = 1:length(R2)
25     fprintf('%0.4f/(s - (%0.4f)) + ', R2(i), P2(i));
26 end
27 if ~isempty(K2)
28     for i = 1:length(K2)
29         fprintf('%0.4f*s^%d + ', K2(i), length(K2)-i);
30     end
31 end
32 fprintf('\b\b \n'); % Remove o último ' + '
33
34 N = 6*[1 34];
35 D = [1 10 34 0];
36 [R3,P3,K3] = residue(N,D);
37
38 disp('Questão 5.c:');
39 for i = 1:length(R3)
40     fprintf('%0.4f/(s - (%0.4f)) + ', R3(i), P3(i));
41 end
42 if ~isempty(K3)
43     for i = 1:length(K3)
44         fprintf('%0.4f*s^%d + ', K3(i), length(K3)-i);
45     end
46 end
47 fprintf('\b\b \n'); % Remove o último ' + '
```

```
Command Window
Questão 5.a:
-1.4214/(s - (-1.4600)) + -30.1909/(s - (-0.1300)) + 31.6122/(s - (0.0000))
Questão 5.b:
0.0000/(s - (-1.0000)) + 2.0000/(s - (-1.0000)) + 1.0000*s^0
Questão 5.c:
-3.0000/(s - (-5.0000)) + -3.0000/(s - (-5.0000)) + 6.0000/(s - (0.0000))
```

Figura 5: Resultado da Questão 5

Questão 6

```
1 %% 6
2
3 t = linspace(0, 5, 1000);
4 x1 = cos(t).*sin(20*t);
5 x2 = cos(t);
6 x3 = sin(20*t);
7
8 figure;
9 plot(t,x1,'b');
10 hold on;
11 plot(t,x2,'r');
12 plot(t,x3,'g');
13 legend('x1','x2','x3');
14 title('Questão 6');
15 xlabel('Tempo[s]');
16 ylabel('Amplitude');
```

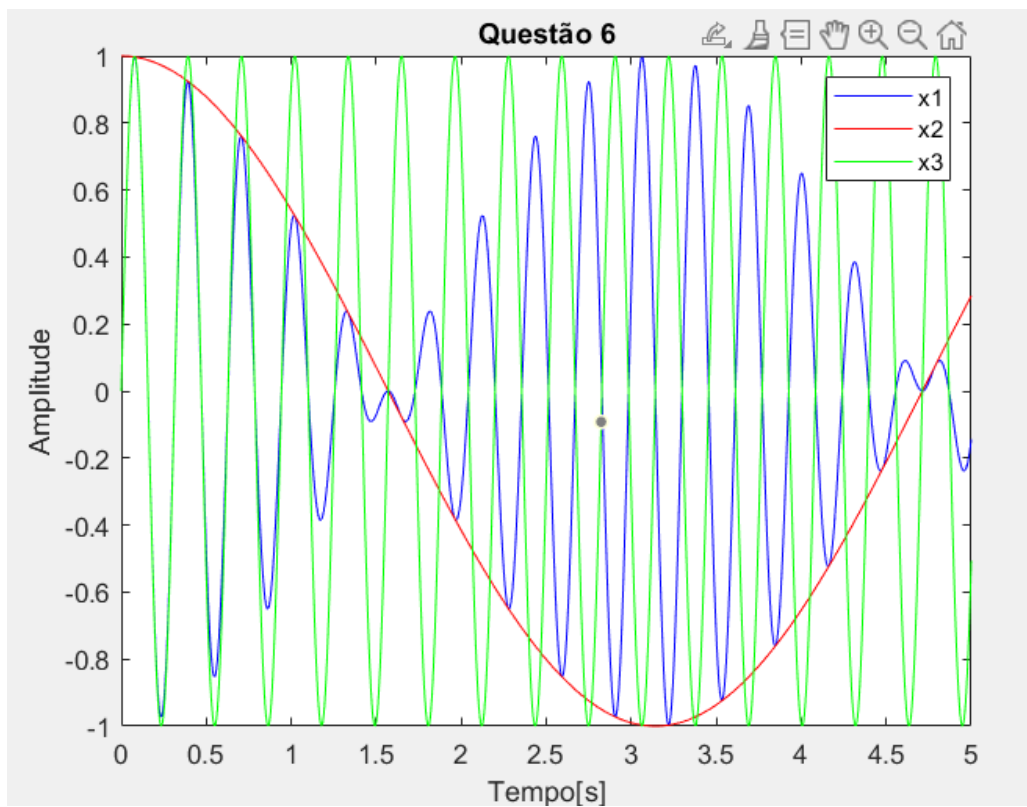


Figura 6: Resultado da Questão 6

Questão 7

```

1 %% 7
2
3 t = linspace(0, 5, 1000);
4 w = 5;
5 f = -3*cos(w*t)+4*sin(w*t);
6 A = sqrt((-3)^2+4^2);
7 phi = atan2(4,-3);
8 f_eq = A*cos(w*t+phi);
9 figure;
10 plot(t,f,'k');
11 title('Questão 7');
12 xlabel('Tempo[s]');
13 ylabel('Amplitude');
14 figure;
15 plot(t,f_eq,'r');
16 title('Questão 7 - função equivalente');
17 xlabel('Tempo[s]');
18 ylabel('Amplitude');

```

A função equivalente de $f = -3 \cos(\omega_0 t) + 4 \sin(\omega_0 t)$ é $f_{eq} = A \cos(\omega_0 t + \phi)$; sendo $A = \sqrt{(-3)^2 + 4^2}$ e $\phi = \text{atan}(\frac{4}{-3})$. Assim, $f_{eq} = 5 \cos(\omega_0 t + 2.2143)$.

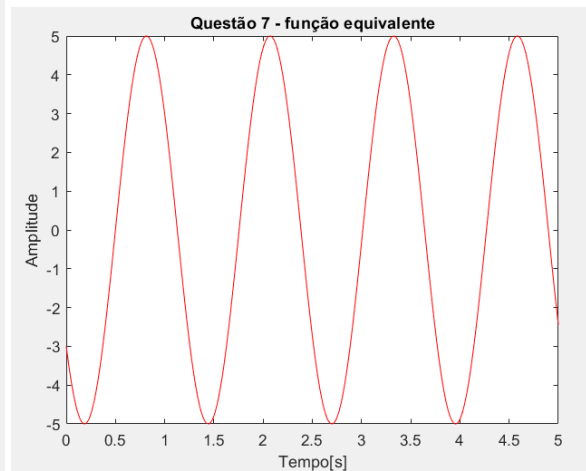
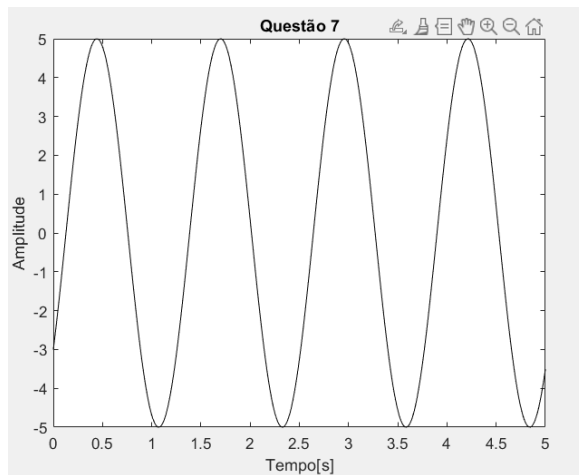


Figura 7: Resultados da Questão 7

Questão 8

```
1 %% 8
2 disp('Questão 8');
3 A = [1 1 6; 5 -2 1; -8 2 -3];
4 B = [2 9; -5 -1; 9 2];
5
6 disp('A');
7 if (size(A,1) == size(A, 2))
8     disp('É uma matriz quadrada');
9 else
10     disp('Não é uma matriz quadrada');
11 end
12
13 disp('Elementos com valor 2:');
14 for i = 1:size(A,1)
15     for j = 1:size(A,2)
16         if(A(i,j) == 2)
17             a = [i,j];
18             disp(a);
19         end
20     end
21 end
22
23 disp('Elementos com valores negativos:');
24 for i = 1:size(A,1)
25     for j = 1:size(A,2)
26         if(A(i,j) < 0)
27             a = [i,j];
28             disp(a);
29         end
30     end
31 end
32
33 disp('B');
34 if (size(B,1) == size(B, 2))
35     disp('É uma matriz quadrada');
36 else
37     disp('Não é uma matriz quadrada');
38 end
39
40 disp('Elementos com valor 2:');
41 for i = 1:size(B,1)
42     for j = 1:size(B,2)
43         if(B(i,j) == 2)
44             a = [i,j];
45             disp(a);
46         end
47     end
48 end
49
50 disp('Elementos com valores negativos:');
51 for i = 1:size(B,1)
52     for j = 1:size(B,2)
53         if(B(i,j) < 0)
54             a = [i,j];
55             disp(a);
56         end
57     end
```

```
Command Window

Questão 8
A
É uma matriz quadrada
Elementos com valor 2:
    3    2

Elementos com valores negativos:
    2    2

    3    1

    3    3

B
Não é uma matriz quadrada
Elementos com valor 2:
    1    1

    3    2

Elementos com valores negativos:
    2    1

    2    2
```

Figura 8: Resultado da Questão 8

Questão 9

```
1 %% 9
2
3 syms a b c d;
4 M = [a b; c d];
5 d = det(M);
6 disp('Questão 9');
7 fprintf('Determinante: %s\n', d);
8 I = inv(M);
9 disp('Inversa:');
10 disp(I);
11 t = trace(M);
12 fprintf('Traço: %s\n', t);
```

Command Window

Questão 9

Determinante: $a*d - b*c$

Inversa:

$\begin{bmatrix} d/(a*d - b*c), & -b/(a*d - b*c) \end{bmatrix}$

$\begin{bmatrix} -c/(a*d - b*c), & a/(a*d - b*c) \end{bmatrix}$

Traço: $a + d$

Figura 9: Resultado da Questão 9

Questão 10

```
1 %% 10
2
3 disp('Questão 10');
4 syms s;
5 f = s^4 + 5*s^3 - s^2 + 3*s + 2;
6 df = diff(f,s);
7 df2 = diff(f,s,2);
8 disp('Primeira derivada:');
9 pretty(df);
10 disp('Segunda derivada:');
11 pretty(df2);
```

Command Window

Questão 10

Primeira derivada:

$$4s^3 + 15s^2 - 2s + 3$$

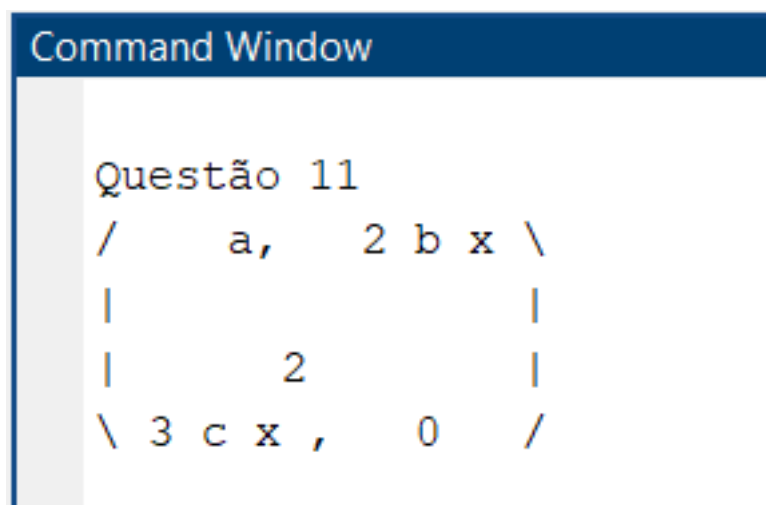
Segunda derivada:

$$12s^2 + 30s - 2$$

Figura 10: Resultado da Questão 10

Questão 11

```
1 %% 11
2
3 syms a b c d x y;
4 M = [a*x b*x^2; c*x^3 d*y];
5 disp('Questão 11');
6 df = diff(M,x);
7 pretty(df);
8
9 hold off;
```



Command Window

Questão 11

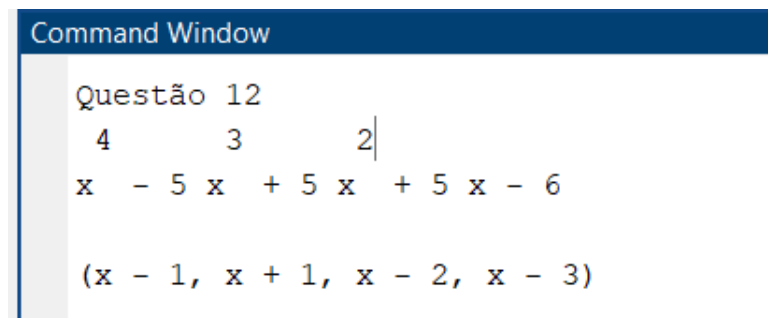
$$\begin{array}{ccccc} / & a, & 2 b x & \backslash \\ | & & & | \\ | & 2 & & | \\ \backslash & 3 c x, & 0 & / \end{array}$$

Figura 11: Resultado da Questão 11

Questão 12

```
1 %% 12
2
3 syms x;
4 p = (x^2-1)*(x-2)*(x-3);
5 ex = expand(p);
6 fac = factor(p);
7 disp('Questão 12');
8 pretty(ex);
9 pretty(fac);
```

O comando *expand()* é usado para aplicar as propriedades distributivas e os produtos notáveis em funções simbólicas. Já o comando *factor()* é usado para fatorar expressões simbólicas, reescrevendo-as em produtos de termos mais simples.



```
Command Window

Questão 12
      4      3      2
x  - 5 x  + 5 x  + 5 x - 6

(x - 1, x + 1, x - 2, x - 3)
```

Figura 12: Resultado da Questão 12