TPC 2 – Introdução à Programação em MATLAB

1.

```
syms x1 x2 x3
eqn1 = 2*x1 + 3*x2 == 0;
eqn2 = 3*x2 + 4*x3 == -1;
eqn3 = x1 + x2 + x3 == -2;
sol = solve([eqn1, eqn2, eqn3], [x1, x2, x3]);
x1Sol = sol.x1
x2Sol = sol.x2
x3Sol = sol.x3
```

Output:

x1Sol = (sym) $\frac{-21}{10}$

x2Sol = (sym) $\frac{7}{5}$

x3Sol = (sym) $\frac{-13}{10}$

2.

a)
```
p = [1, 0, 1, -1];
roots(p)
```

Output:
```
ans =
 -0.3412 + 1.1615i
 -0.3412 – 1.1615i
 0.6823 + 0i
```

b)
```
p = [1, 4, 0, 0, 2];
roots(p)
```

Output:
```
ans =
-3.9680 +      0i
-0.8604 +      0i
 0.4142 + 0.6436i
 0.4142 – 0.6436i
```

c)
```
p = [1, 0, -3, 2, -1, 4];
roots(p)
```

Output:
```
ans = -2.1366 +      0i
       1.3314 + 0.5316i
       1.3314 - 0.5316i
      -0.2631 + 0.9174i
      -0.2631 - 0.9174i
```

3.

a) A = [1 0 0; 0 1 3; 0 1 4];
[vetoresProprios, valoresProprios] = eig(A)

Output:

```
vetoresProprios =
         0         0    1.0000
    0.6205    0.9669         0
    0.7842   -0.2550         0

valoresProprios =
Diagonal Matrix

    4.7913         0         0
         0    0.2087         0
         0         0    1.0000
```

b) A = [1 2 3; 4 5 6; 7 8 9];
[vetoresProprios, valoresProprios] = eig(A)

Output:

```
vetoresProprios =
   -0.231971  -0.785830   0.408248
   -0.525322  -0.086751  -0.816497
   -0.818673   0.612328   0.408248

valoresProprios =
Diagonal Matrix

   1.6117e+01         0         0
         0   -1.1168e+00         0
         0         0   -1.3037e-15
```

c) A = [1 1 0; 0 1 0; 0 0 2];
[vetoresProprios, valoresProprios] = eig(A)

Output:

```
vetoresProprios =
    1.0000   -1.0000         0
         0    0.0000         0
         0         0    1.0000

valoresProprios =
Diagonal Matrix

     1     0     0
     0     1     0
     0     0     2
```

4.

a)
```
function sum = iterativeCubedSum(n)
    sum = 0;
    for i = 1:n
        sum += i.^3;
    endfor
```

Output (para n = 3): ans = 36

b)
```
function sum = recurseCubedSum(n)
    if n < 1
        sum = 0;
        return;
    endif

    if n == 0
        sum = 0;
    else
        sum = n.^3;
        sum += recurseCubedSum(n -1);
    endif
```

Output (para n = 5): ans = 225