Solution of Laboratory Work 5

Exercise 1:

```
clc;

clear;

// 1. Create p(s) = 3 + 2s + s^2 and display it

--> s = poly(0, 's');

--> p = 3 + 2*s + s^2;

--> disp("p(s) =", p);

"p(s) ="

  3 +2s +s^2

// 2. Define a rational fraction q(s) = p(s)/ (1 + 2s)

--> q = p/(1 + 2*s);

--> disp("q(s) =",q);

  "q(s) ="

   3 +2s +s^2

   ----------

     1 +2s

// 3. Create a polynomial variable x and define f(x) =
x^3 - 4x + 2

--> x = poly(0, 'x');
```

```
--> f = x^3 - 4*x + 2;

--> disp("f(x) =",f);

"f(x) ="

  2 -4x +x^3

// 4. Define a polynomial from its roots [-2, 0, 2]

--> r = poly([-2, 0, 2], 'x', 'roots');

--> disp("Polynomial from roots =", r);

"Polynomial from roots ="

  -4x +x^3

// 5. Define a polynomial from coefficients [2, -3, 1,
4] (ascending degree)

--> p2 = poly([2, -3, 1, 4], 'x', 'coeff');

--> disp("Polynomial from coefficients =",p2);

  "Polynomial from coefficients ="

  2 -3x +x^2 +4x^3

//  6.  Compute  the  characteristic  polynomial  of  a
matrix

--> A = [1 2; 3 4];

--> char_poly = poly(A, 'x');

-->   disp("Characteristic   polynomial   of   A   =",
char_poly);

"Characteristic polynomial of A ="

  -2 -5x +x^2
```

Exercise 2:

```
clc;

// 1. Define P(x) = x^3 - 5x^2 + 2x + 8 and find its
zeros

--> x = poly(0, 'x');

--> P = x^3 - 5*x^2 + 2*x + 8;

--> zP = roots(P);

--> disp("Zeros of P(x) =", zP);

  "Zeros of P(x) ="

   4. + 0.i

   2. + 0.i

  -1. + 0.i

// 2. Verify one of the zeros (first one)

--> test_val = horner(P, zP(1));

disp("Verification of first zero:", test_val);

 "Verification of first zero:"

  -6.217D-14

// 3. Define R(x) = x^4 - 2x^3 - 7x^2 + 8x + 12 and
compute its zeros

--> R = x^4 - 2*x^3 - 7*x^2 + 8*x + 12;

--> zR = roots(R);

--> disp("Zeros of R(x) =", zR);

  "Zeros of R(x) ="

   3. + 0.i
```

```
   2. + 0.i

  -2. + 0.i

  -1. + 0.i
```

Exercise 3:

```
clc;

// 1. Define p1(s) and p2(s)

--> s = poly(0, 's');

--> p1 = s^3 - 2*s + 1;

--> p2 = 2*s^2 + s + 3;

// Operations

--> disp("p1 + p2 =", p1 + p2);

  "p1 + p2 ="

  4 -s +2s^2 +s^3

--> disp("p1 - p2 =", p1 - p2);

 "p1 - p2 ="

  -2 -3s -2s^2 +s^3

--> disp("p1 * p2 =",p1 * p2);

  "p1 * p2 ="

  3 -5s -s^3 +s^4 +2s^5

--> disp("p1 / p2 =" , p1 / p2);

  "p1 / p2 ="
```

```
   1 -2s +s^3
   ----------
   3 +s +2s^2
--> disp("p1^3 =",p1^3);
"p1^3 ="
  1 -6s +12s^2 -5s^3 -12s^4 +12s^5 +3s^6 -6s^7 +s^9
// 2. Euclidean division of p1 by p2
--> [q, r] = pdiv(p1, p2);
--> disp("Quotient =", q);
"Quotient ="
  1.75 -3.25s
--> disp("Remainder =", r);
"Remainder ="
  -0.25 +0.5s
// 3. Increasing powers division (use pfactors)
--> disp("Increasing powers division of p1 by p2 =",
pdiv(p1, p2));
"Increasing powers division of p1 by p2 ="
  -0.25 +0.5s
```

Exercise 4:

```
clc;
// Define p(s) and q(s)
--> s = poly(0, 's');
--> p = s^4 - 3*s^3 + 2*s - 1;
--> q = s^3 - s^2 + s - 1;
// 1. Degree of p
--> deg_p = degree(p);
--> disp("Degree of p =", deg_p);
"Degree of p ="
   4.
// 2. Derivative of p
--> dp = derivat(p);
--> disp("Derivative of p =", dp);
"Derivative of p ="
  2 -9s^2 +4s^3
// 3. Coefficients of p
--> coeff_p = coeff(p);
--> disp("Coefficients of p =", coeff_p);
"Coefficients of p ="
  -1.   2.   0.  -3.   1.
```

```
// 4. Variable name

--> vname = varn(p);

--> disp("Variable name =", vname);

"Variable name ="

  "s"

// 5. Factorize p and q

-> disp("Factorization of p =", factors(p));

  "Factorization of p ="

  (1) = -2.7889732+%s

  (2) = 0.8947036+%s

  (3) = 0.4007527-1.1057304*%s+%s^2

--> disp( "Factorization of q =", factors(q));

"Factorization of q ="

  (1) = 1+8.882D-16*%s+%s^2

  (2) = -1+%s

// 6. LCM and GCD

--> g = bezout(p, q);

--> l = (p * q) / g;

--> disp("GCD(p, q) =", g);

  "GCD(p, q) ="

               1

--> disp("LCM(p, q) =",l);
```

```
  "LCM(p, q) ="

   1 -3s +3s^2 -2s^4 +4s^5 -4s^6 +s^7

   ---------------------------------

                    1

// 7. Evaluate p(s) for s = 1.5

--> val = horner(p, 1.5);

--> disp("p(1.5) =", val);

  "p(1.5) ="

  -3.0625

// 8. Simplify rational fraction r = p/q

--> r = p / q;

--> r_simplified = simp(r);

-->   disp("Simplified   rational   fraction   r   =",
r_simplified);

  "Simplified rational fraction r ="

   -1 +2s -3s^3 +s^4

   -----------------

    -1 +s -s^2 +s^3
```