

```

> restart:
> DIVIDE := proc(param_A::polynom, param_B::polynom)
    local X, x1, A, B, Q, R, N, M, LA, LB, T;
    A,B := param_A, param_B;

    if B = 0 then
        return 'FAIL';
    fi:

    X := indets(A) union indets(B):

    if nops(X) = 0 then
        if type(A, integer) and type(B, integer) then
            if B <> 0 and (irem(A, B) = 0) then
                return A / B;
            else
                return 'FAIL';
            fi:
        else
            return 'FAIL';
        fi:
    fi:

    x1 := sort([op(X)])[1];
    Q := 0;
    R := A;
    N := degree(R, x1);
    M := degree(B, x1);

    while (R <> 0) and (N >= M) do

        LA := lcoeff(R, x1);
        LB := lcoeff(B, x1);

        T := DIVIDE(LA, LB);
        if T = 'FAIL' then
            return 'FAIL';
        fi:

        T := T * x1^(N - M);
        Q := Q + T;
        R := expand(R - T*B);
        N := degree(R, x1);
    od:

    if R = 0 then
        return Q;
    else

```

```

        return 'FAIL';
    fi;
end proc;

```

```

> printf("\nInput Polynomials:\n"):

```

```

A := (6*y^2 - 5*y*z + z^2)*x^2 + (7*y^2*z - 3*y*z^2)*x + 2*y^2*z^2;
B := (2*y-z)*x+y*z;
C := expand(A*B);

```

Input Polynomials:

$$\begin{aligned}
 A &:= (6y^2 - 5yz + z^2)x^2 + (7y^2z - 3yz^2)x + 2y^2z^2 \\
 B &:= (2y - z)x + yz \\
 C &:= 12x^3y^3 - 16x^3y^2z + 7x^3yz^2 - x^3z^3 + 20x^2y^3z - 18x^2y^2z^2 + 4x^2yz^3 + 11xy^3z^2 \\
 &\quad - 5xy^2z^3 + 2y^3z^3
 \end{aligned} \tag{1}$$

```

> Q_1 := DIVIDE(A,B);
Q_2 := DIVIDE(A+x,B);
Q_3 := DIVIDE(A+2,B);
Q_4 := DIVIDE(C,B);

```

$$Q_1 := (3y - z)x + 2yz$$

$$Q_2 := \text{FAIL}$$

$$Q_3 := \text{FAIL}$$

$$Q_4 := (6y^2 - 5yz + z^2)x^2 + (7y^2z - 3yz^2)x + 2y^2z^2 \tag{2}$$

```

> printf("\nDivision Checks:\n"):

```

```

divide(A, B);
divide(A+x, B);
divide(A+2, B);
divide(C, B);

```

Division Checks:

true

false

false

true

(3)