```
> restart:
> with(LinearAlgebra): with(PolynomialTools); with(ListTools);
```

[*AnnihilatingPolynomial*, *CoefficientList*, *CoefficientVector*, *FromCoefficientList*, *FromCoefficientVector*, *GcdFreeBasis*, *GreatestFactorialFactorization*, *Homogenize*, *Hurwitz*, *IsHomogeneous*, *IsSelfReciprocal*, *MinimalPolynomial*, *PDEToPolynomial*, *PolynomialToPDE*, *ShiftEquivalent*, *ShiftlessDecomposition*, *Shorten*, *Shorter*, *Sort*, *Split*, *Splits*, *SquareFreePart*, *Translate*]

[*BinaryPlace*, *BinarySearch*, *Categorize*, *Classify*, *Collect*, *DotProduct*, *Enumerate*, *FindMaximalElement*, *FindMinimalElement*, *FindRepetitions*, *Flatten*, *FlattenOnce*, *Group*, *Interleave*, *Join*, *JoinSequence*, *LengthSplit*, *MakeUnique*, *Occurrences*, *Pad*, *PartialSums*, *Reverse*, *Rotate*, *Search*, *SearchAll*, *SelectFirst*, *SelectLast*, *Sorted*, *Split*, *Transpose*] **(1)**

```
> m := 5: f := sum('a'['i']*'x'^'i', 'i'=0..m);
```

$$f := a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \quad \textbf{(2)}$$

```
> n := 4: g := sum('b'['i']*'x'^'i', 'i'=0..n);
```

$$g := b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 \quad \textbf{(3)}$$

```
> S := SylvesterMatrix(f,g,'x');
```

$$S := \begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 & 0 & 0 \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 & 0 \\ 0 & 0 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & 0 & 0 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ b_4 & b_3 & b_2 & b_1 & b_0 & 0 & 0 & 0 & 0 \\ 0 & b_4 & b_3 & b_2 & b_1 & b_0 & 0 & 0 & 0 \\ 0 & 0 & b_4 & b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & 0 & 0 & b_4 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & 0 & 0 & b_4 & b_3 & b_2 & b_1 & b_0 \end{bmatrix} \quad \textbf{(4)}$$

```
> f_y := expand(product(x-'y'['i'],'i'=1..m));
```

$$f_y := x^5 - x^4 y_1 - x^4 y_2 - x^4 y_3 - x^4 y_4 - x^4 y_5 + x^3 y_1 y_2 + x^3 y_1 y_3 + x^3 y_1 y_4 + x^3 y_1 y_5 + x^3 y_2 y_3 + x^3 y_2 y_4 + x^3 y_2 y_5 + x^3 y_3 y_4 + x^3 y_3 y_5 + x^3 y_4 y_5 - x^2 y_1 y_2 y_3 - x^2 y_1 y_2 y_4 - x^2 y_1 y_2 y_5 - x^2 y_1 y_3 y_4 - x^2 y_1 y_3 y_5 - x^2 y_1 y_4 y_5 - x^2 y_2 y_3 y_4 - x^2 y_2 y_3 y_5 - x^2 y_2 y_4 y_5 - x^2 y_3 y_4 y_5 + x y_1 y_2 y_3 y_4 + x y_1 y_2 y_3 y_5 + x y_1 y_2 y_4 y_5 + x y_1 y_3 y_4 y_5 + x y_2 y_3 y_4 y_5 - y_1 y_2 y_3 y_4 y_5 \quad \textbf{(5)}$$

```
> coeff_L:=CoefficientList(f_y,x);
```

$$coeff_L := [-y_1 y_2 y_3 y_4 y_5, y_1 y_2 y_3 y_4 + y_1 y_2 y_3 y_5 + y_1 y_2 y_4 y_5 + y_1 y_3 y_4 y_5 + y_2 y_3 y_4 y_5, -y_1 y_2 y_3 - y_1 y_2 y_4 - y_1 y_2 y_5 - y_1 y_3 y_4 - y_1 y_3 y_5 - y_1 y_4 y_5 - y_2 y_3 y_4 - y_2 y_3 y_5 - y_2 y_4 y_5 - y_3 y_4 y_5, y_1 y_2 + y_1 y_3 + y_1 y_4 + y_1 y_5 + y_2 y_3 + y_2 y_4 + y_2 y_5 + y_3 y_4 + y_3 y_5 + y_4 y_5, -y_1 - y_2 - y_3 - y_4 - y_5, 1] \quad \textbf{(6)}$$

```
> subs_L:=[]:
  for i from 1 to nops(coeff_L) do
```

```
      subs_L := [op(subs_L),a[i-1] = coeff_L[i]];
  od: subs_L;
```

$$\left[a_0 = -y_1 y_2 y_3 y_4 y_5,\ a_1 = y_1 y_2 y_3 y_4 + y_1 y_2 y_3 y_5 + y_1 y_2 y_4 y_5 + y_1 y_3 y_4 y_5 + y_2 y_3 y_4 y_5,\ a_2 = -y_1 y_2 y_3 - y_1 y_2 y_4 - y_1 y_2 y_5 - y_1 y_3 y_4 - y_1 y_3 y_5 - y_1 y_4 y_5 - y_2 y_3 y_4 - y_2 y_3 y_5 - y_2 y_4 y_5 - y_3 y_4 y_5,\ a_3 = y_1 y_2 + y_1 y_3 + y_1 y_4 + y_1 y_5 + y_2 y_3 + y_2 y_4 + y_2 y_5 + y_3 y_4 + y_3 y_5 + y_4 y_5,\ a_4 = -y_1 - y_2 - y_3 - y_4 - y_5,\ a_5 = 1\right] \tag{7}$$

```
> factor(Determinant(map(x -> subs(subs_L,x),S)));
```

$$\left(b_4 y_5^4 + b_3 y_5^3 + b_2 y_5^2 + b_1 y_5 + b_0\right)\left(b_4 y_4^4 + b_3 y_4^3 + b_2 y_4^2 + b_1 y_4 + b_0\right)\left(b_4 y_3^4 + b_3 y_3^3 + b_2 y_3^2 + b_1 y_3 + b_0\right)\left(b_4 y_2^4 + b_3 y_2^3 + b_2 y_2^2 + b_1 y_2 + b_0\right)\left(b_4 y_1^4 + b_3 y_1^3 + b_2 y_1^2 + b_1 y_1 + b_0\right) \tag{8}$$

```
> quo(b[4]^3*f,g,'x','r');
```

$$a_5 b_4^2 x + \left(a_4 b_4 - a_5 b_3\right) b_4 \tag{9}$$

```
> r;
```

$$\left(a_3 b_4^2 - a_4 b_3 b_4 - a_5 b_2 b_4 + a_5 b_3^2\right) b_4 x^3 + \left(a_2 b_4^2 - a_4 b_2 b_4 - a_5 b_1 b_4 + a_5 b_2 b_3\right) b_4 x^2 + \left(a_1 b_4^2 - a_4 b_1 b_4 - a_5 b_0 b_4 + a_5 b_1 b_3\right) b_4 x + \left(a_0 b_4^2 - a_4 b_0 b_4 + a_5 b_0 b_3\right) b_4 \tag{10}$$

```
> gcd(f,g);
```

$$1 \tag{11}$$

```
> w1 := sum('u'['i']*'x'^'i','i'=0..n-1);
  w2 := sum('v'['i']*'x'^'i','i'=0..m-1);
```

$$w1 := u_3 x^3 + u_2 x^2 + u_1 x + u_0$$
$$w2 := v_4 x^4 + v_3 x^3 + v_2 x^2 + v_1 x + v_0 \tag{12}$$

```
> w := Vector['row']([Reverse(CoefficientList(w1,'x')), Reverse
  (CoefficientList(w2,'x'))]);
```

$$w := \begin{bmatrix} u_3 & u_2 & u_1 & u_0 & v_4 & v_3 & v_2 & v_1 & v_0 \end{bmatrix} \tag{13}$$

```
> wtimesStrans:=convert(w . S, Vector['column']);
```

$$wtimesStrans := \begin{bmatrix} u_3 a_5 + v_4 b_4 \\ u_3 a_4 + u_2 a_5 + v_4 b_3 + v_3 b_4 \\ u_3 a_3 + u_2 a_4 + u_1 a_5 + v_4 b_2 + v_3 b_3 + v_2 b_4 \\ u_3 a_2 + u_2 a_3 + u_1 a_4 + u_0 a_5 + v_4 b_1 + v_3 b_2 + v_2 b_3 + v_1 b_4 \\ u_3 a_1 + u_2 a_2 + u_1 a_3 + u_0 a_4 + v_4 b_0 + v_3 b_1 + v_2 b_2 + v_1 b_3 + v_0 b_4 \\ u_3 a_0 + u_2 a_1 + u_1 a_2 + u_0 a_3 + v_3 b_0 + v_2 b_1 + v_1 b_2 + v_0 b_3 \\ u_2 a_0 + u_1 a_1 + u_0 a_2 + v_2 b_0 + v_1 b_1 + v_0 b_2 \\ u_1 a_0 + u_0 a_1 + v_1 b_0 + v_0 b_1 \\ u_0 a_0 + v_0 b_0 \end{bmatrix} \tag{14}$$

```
> coeffvec:=Vector(Reverse(CoefficientList(expand(w1*f+w2*g),'x')))
```

```
;
```

$$coeffvec := \begin{bmatrix} u_3 a_5 + v_4 b_4 \\ u_3 a_4 + u_2 a_5 + v_4 b_3 + v_3 b_4 \\ u_3 a_3 + u_2 a_4 + u_1 a_5 + v_4 b_2 + v_3 b_3 + v_2 b_4 \\ u_3 a_2 + u_2 a_3 + u_1 a_4 + u_0 a_5 + v_4 b_1 + v_3 b_2 + v_2 b_3 + v_1 b_4 \\ u_3 a_1 + u_2 a_2 + u_1 a_3 + u_0 a_4 + v_4 b_0 + v_3 b_1 + v_2 b_2 + v_1 b_3 + v_0 b_4 \\ u_3 a_0 + u_2 a_1 + u_1 a_2 + u_0 a_3 + v_3 b_0 + v_2 b_1 + v_1 b_2 + v_0 b_3 \\ u_2 a_0 + u_1 a_1 + u_0 a_2 + v_2 b_0 + v_1 b_1 + v_0 b_2 \\ u_1 a_0 + u_0 a_1 + v_1 b_0 + v_0 b_1 \\ u_0 a_0 + v_0 b_0 \end{bmatrix} \quad (15)$$

```
> wtimesStrans-coeffvec;
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (16)$$

```
> read("subres.mpl");
```

$$F := x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5$$

$$G := 3x^6 + 5x^4 - 4x^2 - 9x + 21$$

```
subres := proc(F, G, x, j)
    local U, V, g, h, delta, R, ell, g2, h2, SU, SV, TU, TV, Q, tmp;
    SU := 1;
    TU := 0;
    SV := 0;
    TV := 1;
    U := F;
    V := G;
    g := 1;
    h := 1;
    delta := degree(U, x) - degree(V, x);
    while j < degree(V, x) do
        ell := lcoeff(V, x);
        R := rem(ell^(delta + 1) * U, V, x);
        userinfo(3, 'subres', "deg(R)=", degree(R, x));
```

```
        Q := quo(ell^(delta + 1) * U, V, x);
        if R = 0 then RETURN(0) end if;
        U := V;
        V := (-1)^(delta + 1) * R / (g * h^delta);
        userinfo(3, 'subres', "V=", V);
        tmp := SU;
        SU := SV;
        SV := expand((-1)^(delta + 1) * (ell^(delta + 1) * tmp - Q * SU) / (g * h^delta));
        tmp := TU;
        TU := TV;
        TV := expand((-1)^(delta + 1) * (ell^(delta + 1) * tmp - Q * TU) / (g * h^delta));
        userinfo(3, 'subres', "SV=", SV, "TV=", TV, ",  SV * F + TV * G = ", expand(SV * F
         + TV * G));
        g := lcoeff(U, x);
        h := h^(1 - delta) * g^delta;
        g2 := lcoeff(V, x);
        h2 := h^(1 - delta) * g2^delta;
        if degree(V, x) < degree(U, x) - 1 then
            userinfo(3, 'subres', "H=", expand(h2 * V / g2))
        end if;
        delta := degree(U, x) - degree(V, x)
    end do;
    g := lcoeff(V, x);
    h := h^(1 - delta) * g^delta;
    if degree(V, x) = j then RETURN(h * V / g) end if;
    if degree(U, x) - 1 = j then RETURN(V) end if;
    RETURN(0)
end proc
toedet := proc(row1, col1)
    local i, n, x, Tau, S;
    n := linalg:-vectdim(row1);
    Tau := 0;
    x := 'x';
    for i from n by -1 to 1 do Tau := Tau + col1[i] * x^(n - i) end do;
    for i from 2 to n do Tau := Tau + row1[i] * x^(i + n - 2) end do;
    print(x^(2 * n - 1));
    print(Tau);
    S := subres(x^(2 * n - 1), Tau, x, n - 1);
    if degree(S, x) < n - 1 then RETURN(0) end if;
    RETURN(lcoeff(S, x))
end proc
subresp := proc(F, G, x, j, p)
    local U, V, g, h, delta, R, ell, i, GG, gg, hh, del;
    print(`F = `, F, `G = `, G);
    print(`Fb = `, F mod p, `Gb = `, G mod p);
    U := F mod p;
    V := G mod p;
    g := 1;
    h := 1;
    print(`V = `, V);
    delta := degree(U, x) - degree(V, x);
```

```
    print(`delta = `, delta);
    GG[1] := U;
    GG[2] := V;
    del[1] := degree(GG[1], x) - degree(GG[2], x);
    gg[1] := lcoeff(GG[1], x);
    gg[2] := lcoeff(GG[2], x);
    hh[2] := gg[2]^del[1];
    i := 2;
    while j < degree(V, x) do
        i := i + 1;
        print(`i = `, i);
        if i <= 3 then
            GG[i] := (-1)^(del[i - 2] + 1) * Rem(gg[i - 1]^(del[i - 2] + 1) * GG
            [i - 2] mod p, GG[i - 1], x) mod p
        else
            GG[i] := (-1)^(del[i - 2] + 1) * Rem(gg[i - 1]^(del[i - 2] + 1) * GG
            [i - 2] mod p, GG[i - 1], x) / (gg[i - 2] * hh[i - 2]^del[i - 2]) mod p
        end if;
        del[i - 1] := degree(GG[i - 1], x) - degree(GG[i], x);
        gg[i] := lcoeff(GG[i], x);
        hh[i] := gg[i]^del[i - 1] * hh[i - 1]^(1 - del[i - 1]) mod p;
        ell := lcoeff(V, x);
        R := Rem(ell^(delta + 1) * U mod p, V, x) mod p;
        if R = 0 then RETURN(0) end if;
        U := V;
        V := (-1)^(delta + 1) * R / (g * h^delta) mod p;
        g := lcoeff(U, x);
        h := h^(1 - delta) * g^delta mod p;
        print(`g = `, g, `h = `, h);
        print(`h/g * U = `, h * U/g mod p);
        delta := degree(U, x) - degree(V, x);
        print(`delta = `, delta);
        print(`gg[i-1]=`, gg[i - 1], `hh[i-1]=`, hh[i - 1]);
        print(`gg[i]=`, gg[i], `hh[i]=`, hh[i]);
        print(`V=`, V);
        print(`GG=`, GG[i]);
        print(`hh[i]/gg[i]*GG=`, hh[i] * GG[i]/gg[i] mod p);
        print(`subres = `, linalg:-det(subresmat(F, G, x, degree(V, x))) mod p)
    end do;
    g := lcoeff(V, x);
    h := h^(1 - delta) * g^delta mod p;
    print(`g = `, g, `h = `, h);
    if degree(V, x) = j then RETURN(h * V/g mod p) end if;
    if degree(U, x) - 1 = j then RETURN(V) end if;
    RETURN(0)
end proc
toedetp := proc(row1, col1, p)
    local i, n, x, Tau, S;
    n := linalg:-vectdim(row1);
    Tau := 0;
    x := 'x';
```

```
    for i from n by -1 to 1 do Tau := Tau + col1[i]*x^(n-i) mod p end do;
    for i from 2 to n do Tau := Tau + row1[i]*x^(i+n-2) mod p end do;
    print(x^(2*n-1));
    print(Tau);
    S := subresp(x^(2*n-1), Tau, x, n-1, p);
    if degree(S,x) < n-1 then RETURN(0) end if;
    RETURN(lcoeff(S,x))
end proc
subresmat := proc(G,H,x,j)
    local m, ell, syl, rcol1, rcol2, i;
    m := degree(G,x);
    ell := degree(H,x);
    syl := linalg:-sylvester(G,H,x);
    rcol1 := linalg:-matrix(ell-j,1);
    for i from 0 to ell-j-1 do rcol1[i+1,1] := x^(ell-j-1-i)*G end do;
    rcol2 := linalg:-matrix(m-j,1);
    for i from 0 to m-j-1 do rcol2[i+1,1] := x^(m-j-1-i)*H end do;
    linalg:-blockmatrix(2,2,[linalg:-submatrix(syl,1..ell-j,1..ell+m-2*j-1),rcol1,
    linalg:-submatrix(syl,ell+1..ell+m-j,1..ell+m-2*j-1),rcol2])
end proc
lcsubresmat := proc(G,H,x,j)                                              (17)
    local m, ell, syl, rcol1, rcol2, i;
    m := degree(G,x);
    ell := degree(H,x);
    syl := linalg:-sylvester(G,H,x);
    rcol1 := linalg:-matrix(ell-j,1);
    for i from 0 to ell-j-1 do
        rcol1[i+1,1] := coeff(expand(x^(ell-j-1-i)*G),x,j)
    end do;
    rcol2 := linalg:-matrix(m-j,1);
    for i from 0 to m-j-1 do
        rcol2[i+1,1] := coeff(expand(x^(m-j-1-i)*H),x,j)
    end do;
    linalg:-blockmatrix(2,2,[linalg:-submatrix(syl,1..ell-j,1..ell+m-2*j-1),rcol1,
    linalg:-submatrix(syl,ell+1..ell+m-j,1..ell+m-2*j-1),rcol2])
end proc
> SS:=subresmat(f,g,'x',1);
```

(18)

$$SS := \begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & x^2\left(x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0\right) \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 & x\left(x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0\right) \\ 0 & 0 & a_5 & a_4 & a_3 & a_2 & x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0 \\ b_4 & b_3 & b_2 & b_1 & b_0 & 0 & x^3\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & b_4 & b_3 & b_2 & b_1 & b_0 & x^2\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & 0 & b_4 & b_3 & b_2 & b_1 & x\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & 0 & 0 & b_4 & b_3 & b_2 & x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0 \end{bmatrix} \quad (18)$$

```
> SSS:=(convert(SS,Matrix));
```

$$SSS := \begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & x^2\left(x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0\right) \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 & x\left(x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0\right) \\ 0 & 0 & a_5 & a_4 & a_3 & a_2 & x^5 a_5 + x^4 a_4 + x^3 a_3 + x^2 a_2 + x a_1 + a_0 \\ b_4 & b_3 & b_2 & b_1 & b_0 & 0 & x^3\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & b_4 & b_3 & b_2 & b_1 & b_0 & x^2\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & 0 & b_4 & b_3 & b_2 & b_1 & x\left(x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0\right) \\ 0 & 0 & 0 & b_4 & b_3 & b_2 & x^4 b_4 + x^3 b_3 + x^2 b_2 + x b_1 + b_0 \end{bmatrix} \quad (19)$$

```
> map(s -> coeff(expand(s),x,0),SSS);
```

$$\begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_5 & a_4 & a_3 & a_2 & a_1 & 0 \\ 0 & 0 & a_5 & a_4 & a_3 & a_2 & a_0 \\ b_4 & b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_4 & b_3 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_4 & b_3 & b_2 & b_1 & 0 \\ 0 & 0 & 0 & b_4 & b_3 & b_2 & b_0 \end{bmatrix} \quad (20)$$

```
> ?
```

`?` (21)

```
> ?
```

`?` (22)

```
>
```