Bevezetés a matematikába

Járai Antal

Ezek a programok csak szemléltetésre szolgálnak.

▶ 1. Halmazok

▶ 2. Természetes számok

▶ 3. A számfogalom bővítése

▶ 4. Véges halmazok

▶ 5. Végtelen halmazok

▶ 6. Számelmélet

▶ 7. Gráfelmélet

▼ 8. Algebra

▼ 8.1. Csoportok

```
> restart;with(group);
```

[DerivedS, LCS, NormalClosure, RandElement, SnConjugates, Sylow, areconjugate, center, centralizer, core, cosets, cosrep, derived, elements, groupmember, grouporder, inter, invperm, isabelian, isnormal, issubgroup, mulperms, normalizer, orbit, parity, permrep, pres, transgroup] (8.1.1)

▶ *8.1.1. Megjegyzés.*

▶ *->8.1.2. Feladat.*

▶ *8.1.3. Homomorfizmusok.*

▼ *8.1.4. Példa.*

```
> a^(x+y);expand(%);
```

$$a^{x+y}$$
$$a^x\, a^y \qquad (8.1.4.1)$$

```
> 1^2; (-1)^2; 0+0;
```

$$1$$
$$1$$
$$0 \qquad (8.1.4.2)$$

▶ *->8.1.5. Feladat.*

▶ *->8.1.6. Feladat.*

▶ ***8.1.7. Reprezentációk.***

▶ ***8.1.8. Tétel.***

▶ ***8.1.9. Következmény.***

▼ ***8.1.10. Példa.***

```
> solve(x+x=0); solve(x*x=1) assuming real;
```

$$0$$
$$1, -1 \qquad (8.1.10.1)$$

▶ *->8.1.11. Feladat.*

▶ *->8.1.12. Feladat.*

▶ ***8.1.13. Tétel.***

▶ ***8.1.14. Következmény: egyszerűsítési szabály.***

▶ ***8.1.15. Megjegyzés.***

▼ ***8.1.16. Példák.***

```
> solve(z^6=1,z); G:=evalc([%]);
  expand(evalc(1/G[3]));expand(evalc(G[2]*G[3]));
```

$$-1,\, 1,\, -\frac{1}{2}\sqrt{-2+2\,\mathrm{I}\sqrt{3}},\, \frac{1}{2}\sqrt{-2+2\,\mathrm{I}\sqrt{3}},\, -\frac{1}{2}\sqrt{-2-2\,\mathrm{I}\sqrt{3}},\, \frac{1}{2}\sqrt{-2-2\,\mathrm{I}\sqrt{3}}$$

$$G := \left[-1,\, 1,\, -\frac{1}{2}-\frac{1}{2}\,\mathrm{I}\sqrt{3},\, \frac{1}{2}+\frac{1}{2}\,\mathrm{I}\sqrt{3},\, -\frac{1}{2}+\frac{1}{2}\,\mathrm{I}\sqrt{3},\, \frac{1}{2}-\frac{1}{2}\,\mathrm{I}\sqrt{3}\right]$$

$$-\frac{1}{2}+\frac{1}{2}\,\mathrm{I}\sqrt{3}$$

$$-\frac{1}{2}-\frac{1}{2}\,\mathrm{I}\sqrt{3} \qquad (8.1.16.1)$$

```
> undefine('`&*`'); define('`&*`','multilinear','flat',
'identity'=1);

&*(i,i):=-1;&*(j,j):=-1;&*(k,k):=-1;&*(i,j):=k;&*(j,k):=i;
&*(k,i):=j;&*(j,i):=-k;&*(k,j):=-i;&*(i,k):=-j;

(-1*i)&*(-1*k)=i&*k;
```

$$i \,\&^*\, i := -1$$

$$j \,\&^*\, j := -1$$

$$k \,\&^*\, k := -1$$

$$i \,\&^*\, j := k$$

$$j \,\&^*\, k := i$$

$$k \,\&^*\, i := j$$

$$j \,\&^*\, i := -k$$

$$k \,\&^*\, j := -i$$

$$i \,\&^*\, k := -j$$

$$-j = -j \qquad (8.1.16.2)$$

```
> undefine('`&*`'); define('`&*`','flat','orderless',
'identity'=e);
&*(a,b):=c;&*(a,c):=b;&*(b,c):=a;&*(a,a):=e;&*(b,b):=e;&*
(c,c):=e;

e&*a&*c&*b;
```

$$a \,\&^*\, b := c$$

$$a \,\&^*\, c := b$$

$$b \,\&^*\, c := a$$

$$a \,\&^*\, a := e$$

$$b \,\&^*\, b := e$$

$$c \,\&^*\, c := e$$

$$e \qquad (8.1.16.3)$$

8.1.17. Geometriai példák.

```
> D3:=grelgroup({tau,epsilon},{[epsilon,epsilon,epsilon],
[tau,tau],[epsilon,tau,epsilon,tau]});
```

$$D3 := grelgroup\left(\{\tau, \varepsilon\}, \{[\varepsilon, \varepsilon, \varepsilon], [\tau, \tau], [\varepsilon, \tau, \varepsilon, \tau]\}\right) \qquad (8.1.17.1)$$

->8.1.18. Feladat.

->8.1.19. Feladat.

->8.1.20. Feladat.

▶ ***8.1.55. Tétel.***
▶ ***->8.1.56. Feladat.***
▶ ***->8.1.57. Feladat.***
▶ ***->8.1.58. Feladat.***
▶ ***->8.1.59. Feladat.***
▶ ***->8.1.60. Feladat.***
▼ ***->8.1.61. Feladat.***
▼ ***->8.1.62. Feladat.***
▶ ***->8.1.63. Feladat.***
▶ ***8.1.64. Feladat.***
▶ ***8.1.65. Feladat.***
▶ ****8.1.66. Feladat.***
▶ ****8.1.67. Feladat.***
▶ ****8.1.68. Feladat.***
▶ ***->8.1.69. Feladat.***
▶ ***8.1.70. Normálosztó.***
▶ ***8.1.71. Tétel.***
▶ ***8.1.72. Következmény.***
▼ ***8.1.73. Példa.***

```
> H1:=subgrel({x=[tau]},D3); isnormal(H1);
  H2:=subgrel({x=[epsilon]},D3); isnormal(H2);
```

$H1 := subgrel(\{x = [\tau]\}, grelgroup(\{\tau, \varepsilon\}, \{[\varepsilon, \varepsilon, \varepsilon], [\tau, \tau], [\varepsilon, \tau, \varepsilon, \tau]\}))$

$$false$$

$H2 := subgrel(\{x = [\varepsilon]\}, grelgroup(\{\tau, \varepsilon\}, \{[\varepsilon, \varepsilon, \varepsilon], [\tau, \tau], [\varepsilon, \tau, \varepsilon, \tau]\}))$

$$true \qquad (8.1.73.1)$$

▼ ***->8.1.74. Feladat.***
▼ ****8.1.75. Feladat.***
▶ ***8.1.76. Belső automorfizmusok.***
▶ ****8.1.77. Centralizátor és centrum.***
▶ ****8.1.78. Osztályegyenlet.***
▼ ****8.1.79. Példák.***

▶ *8.1.80. Tétel.*
▶ *8.1.81. Következmény.*
▶ *8.1.82. Faktorcsoport.*
▶ *8.1.83. Példák.*
▶ *8.1.84. Homomorfizmus magja.*
▶ *8.1.85. Homomorfizmustétel.*
▶ *->8.1.86. Feladat.*
▶ **8.1.87. Feladat.*
▶ **8.1.88. Feladat.*
▶ *8.1.89. Feladat.*
▶ **8.1.90. Projektív csoportok.*
▶ **8.1.91. Feladat.*
▶ *8.1.92. Direkt szorzat.*
▶ *8.1.93. Véges Abel-csoportok alaptétele.*
▶ **8.1.94. Végesen generált Abel-csoportok alaptétele.*
▶ *->8.1.95. Feladat.*
▶ *8.1.96. Feladat.*
▶ *8.1.97. Feladat.*
▶ *8.1.98. Feladat.*
▶ *8.1.99. Feladat.*
▶ *8.1.100. Feladat.*
▶ *8.1.101. Feladat: diszkrét direkt szorzat.*
▶ *8.1.102. Cayley tétele.*
▼ *8.1.103. Permutációcsoportok.*

```
> convert([3,4,2,1,7,6,5],'disjcyc'); convert(%,'permlist',7)
  ;
```

$$[[1, 3, 2, 4], [5, 7]]$$
$$[3, 4, 2, 1, 7, 6, 5] \quad (8.1.103.1)$$

```
> undefine('`&*`'); `&*`:=(x,y)->mulperms(y,x);

  g:=convert([3,4,2,1,7,6,5],'disjcyc');
  h:=convert([2,5,3,4,1,7,6],'disjcyc');
  f:=g&*h; convert(%,'permlist',7);
```

$$\&* := (x, y) \to group\text{:-}mulperms(y, x)$$
$$g := [[1, 3, 2, 4], [5, 7]]$$
$$h := [[1, 2, 5], [6, 7]]$$

$$f := [[1, 4], [2, 7, 6, 5, 3]]$$

$$[4, 7, 2, 1, 3, 5, 6] \quad (8.1.103.2)$$

```
> invperm(f); invperm(h)&*invperm(g);
```

$$[[1, 4], [2, 3, 5, 6, 7]]$$

$$[[1, 4], [2, 3, 5, 6, 7]] \quad (8.1.103.3)$$

```
> S5:=permgroup(5,{[[1,2]],[[1,2,3,4,5]]}); grouporder(S5);
isabelian(S5);
```

$$S5 := permgroup(5, \{[[1, 2]], [[1, 2, 3, 4, 5]]\})$$

$$120$$

$$false \quad (8.1.103.4)$$

```
> H1:=permgroup(5,{[[1,2,3,4,5]]}); grouporder(H1); isabelian
(H1);
```

$$H1 := permgroup(5, \{[[1, 2, 3, 4, 5]]\})$$

$$5$$

$$true \quad (8.1.103.5)$$

```
> elements(H1);
```

$$\{[\,], [[1, 2, 3, 4, 5]], [[1, 5, 4, 3, 2]], [[1, 4, 2, 5, 3]], [[1, 3, 5, 2, 4]]\} \quad (8.1.103.6)$$

```
> gg:=RandElement(S5); H2:=permgroup(5,{gg}); grouporder(H2);
groupmember([[1,2,3,4,5]],H2); groupmember(gg,H2);
```

$$gg := [[1, 2], [3, 4]]$$

$$H2 := permgroup(5, \{[[1, 2], [3, 4]]\})$$

$$2$$

$$false$$

$$true \quad (8.1.103.7)$$

```
> inter(H1,H2); issubgroup(H1,S5); issubgroup(H1,H2);
isnormal(H1,S5); isnormal(H2,S5); isnormal(H2,H1);
```

$$permgroup(5, \{\})$$

$$true$$

$$false$$

$$true$$

$$true$$

$$false \quad (8.1.103.8)$$

8.1.104. Tétel.

```
> parity(g); parity(h); parity(f);
```

$$1$$

$$-1$$

-1 (8.1.104.1)

8.1.105.Következmény.

8.1.106. Példa.

```
> [[1,3]]&*[[1,2]]&*[[3,4]]; [[1,2]]&*[[2,1]];
```

$$[[1, 2, 3, 4]]$$

$$[\ ]$$ (8.1.106.1)

->8.1.107. Feladat.

->8.1.108. Feladat.

->8.1.109. Feladat.

->8.1.110. Feladat.

->8.1.111. Feladat.

->8.1.112. Feladat.

*8.1.113. Feladat.

->8.1.114. Feladat.

*8.1.115. Feladat.

->8.1.116. Feladat.

8.1.117. Feladat.

8.1.118. Feladat.

*8.1.119. Definíció.

```
> M11:=permgroup(11,{[[1,2,3,4,5,6,7,8,9,10,11]],[[3,7,11,8],
  [4,10,5,6]]});
```

$M11 := permgroup(11, \{[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]], [[3, 7, 11, 8], [4, 10, 5, 6]]\})$ (8.1.119.1)

```
> L:=[1,2];L[1]:=3;L;
```

$$L := [1, 2]$$

$$L_1 := 3$$

$$[3, 2]$$ (8.1.119.2)

```
> grouporder(M11);
```

$$7920$$ (8.1.119.3)

```
> conjugateclasses:=proc(G) local GG,SS,p,i,f;
    SS:=[]; GG:=elements(G);
    for p in GG do
      f:=false;
```

```
      for i to nops(SS) do
        if areconjugate(G,SS[i][1],p) and not p in SS[i] then
          f:=true; SS[i]:=SS[i] union {p}; break;
        fi;
      od;
      if not f then SS:=[op(SS),{p}]; fi;
    od;
    convert(SS,set);
  end;
```

conjugateclasses := **proc**(*G*) (8.1.119.4)

```
    local GG, SS, p, i, f;
    SS := [ ];
    GG := group:-elements(G);
    for p in GG do
        f := false;
        for i to nops(SS) do
            if group:-areconjugate(G, SS[i][1],
            p) and not in(p, SS[i]) then
                f := true;
                SS[i] := union(SS[i], {p});
                break
            end if
        end do;
        if not f then
            SS := [op(SS), {p}]
        end if
    end do;
    convert(SS, set)
end proc
```

```
> conjugateclasses(M11):nops(%);
```

8.1.120. Példa.

```
>
> S4:=permgroup(4,{[[1,2]],[[2,3]],[[3,4]]}); grouporder(S4);
```

$S4 := permgroup(4, \{[[1, 2]], [[3, 4]], [[2, 3]]\})$

$$24 \quad (8.1.120.1)$$

```
> elements(S4); A4:=permgroup(4,select(x->parity(x)=1,%));
  grouporder(A4);
  isnormal(A4,S4); cosets(S4,A4);
```

{[ ], [[1, 2]], [[1, 2], [3, 4]], [[1, 3]], [[3, 4]], [[1, 2, 3]], [[1, 2, 3, 4]], [[2, 3]], [[1, 3, 4, 2]], [[1, 3, 2, 4]], [[1, 4, 3, 2]], [[1, 2, 4, 3]], [[1, 4, 2, 3]], [[2, 3, 4]], [[1, 3, 4]], [[1, 4, 2]], [[1, 3, 2]], [[2, 4]], [[1, 2, 4]], [[2, 4, 3]], [[1, 3], [2, 4]], [[1, 4, 3]], [[1, 4]], [[1, 4], [2, 3]]}

$A4 := permgroup(4, \{[\,], [[1, 2], [3, 4]], [[1, 2, 3]], [[2, 3, 4]], [[1, 3, 4]], [[1, 4, 2]], [[1, 3, 2]], [[1, 2, 4]], [[2, 4, 3]], [[1, 3], [2, 4]], [[1, 4, 3]], [[1, 4], [2, 3]]\})$

12

true

{[ ], [[3, 4]]} (8.1.120.2)

```
> N1:=permgroup(4,{}); grouporder(N1);
```

$N1 := permgroup(4, \{\})$

1 (8.1.120.3)

```
> N2:=permgroup(4,{[[1,2],[3,4]],[[1,3],[2,4]],[[1,4],[2,3]]}
  );
  grouporder(N2); isnormal(N2,A4); cosets(A4,N2);
```

$N2 := permgroup(4, \{[[1, 2], [3, 4]], [[1, 3], [2, 4]], [[1, 4], [2, 3]]\})$

4

true

{[ ], [[2, 3, 4]], [[2, 4, 3]]} (8.1.120.4)

8.1.121. Feladat.
8.1.122. Feladat.
->8.1.123. Feladat.
->8.1.124. Feladat.
8.1.125. Feladat.
8.1.126. Feladat.
8.1.127. Feladat.
8.1.128. Feladat.
8.1.129. Feladat.
8.1.130. Feladat.
*8.1.131. Feladat.
8.1.132. Feladat.
*8.1.133. Feladat.
8.1.134. További feladatok.

8.2. Gyűrűk és testek

```
> restart;
```

8.2.1. Megjegyzés.
**8.2.2. Megjegyzés.*
8.2.3. Példák.

```
> undefine('`&*`'); `&*`:=(x,y)->[x[1]*y[1],x[2]*y[2]];
  [1,0]&*[0,1];
```

$$\&* := (x, y) \to [x_1 y_1, x_2 y_2]$$

$$[0, 0] \tag{8.2.3.1}$$

**8.2.4. Példa.*
->8.2.5. Feladat.
->8.2.6. Feladat.
->8.2.7. Feladat.
**8.2.8. Feladat.*
8.2.9. Feladat.
8.2.10. Feladat.
8.2.11. Feladat.
8.2.12. Feladat.
8.2.13. Feladat.
8.2.14. Feladat.
8.2.15. Homomorfizmusok.
8.2.16. Példák.
->8.2.17. Feladat.
8.2.18. Tétel.
8.2.19. Tétel.

```
> X:={0,1,2,3,4}; map(x->x+x mod 5,X); map(x->x+x+x mod 5,X);
  map(x->x+x+x+x mod 5,X); map(x->x+x+x+x+x mod 5,X);
```

$$X := \{0, 1, 2, 3, 4\}$$

$$\{0, 1, 2, 3, 4\}$$

$$\{0, 1, 2, 3, 4\}$$

$$\{0, 1, 2, 3, 4\}$$

$$\{0\} \tag{8.2.19.1}$$

8.2.53. Homomorfizmus magja.
8.2.54. Homomorfizmus-tétel.
8.2.55. Példa.
->8.2.56. Feladat.
->8.2.57. Feladat.
->8.2.58. Feladat.
->8.2.59. Feladat.
->8.2.60. Feladat.
->8.2.61. Feladat.
->8.2.62. Feladat.
8.2.63. Direkt szorzat.
**8.2.64. Példa.*
8.2.65. Tétel.
8.2.66. Következmény.
8.2.67. Gauss-gyűrűk.
**8.2.68. Példa.*
8.2.69. Euklideszi gyűrűk.
8.2.70. Állítás.
8.2.71. Példa: Gauss-egészek.
8.2.72. Feladat.
8.2.73. Feladat.
8.2.74. Feladat.
8.2.75. Bővített euklideszi algoritmus.

```
> polynomexgcd:=proc(a,b,z) local x0,x1,x2,y0,y1,y2,r0,r1,r2,
  q;
  x0:=1; y0:=0; r0:=a; x1:=0; y1:=1; r1:=b;
  do
    if r1=0 then return [x0,y0,r0] fi;
    q:=quo(r0,r1,z); r2:=expand(r0-q*r1);
    x2:=expand(x0-q*x1); y2:=expand(y0-q*y1);
    r0:=r1; x0:=x1; y0:=y1; r1:=r2; x1:=x2; y1:=y2;
  od; end;
```

polynomexgcd := **proc**(*a*, *b*, *z*) (8.2.75.1)

local *x0*, *x1*, *x2*, *y0*, *y1*, *y2*, *r0*, *r1*, *r2*,
q;
x0 := 1;

```
    y0:= 0;
    r0:= a;
    x1:= 0;
    y1:= 1;
    r1:= b;
    do
        if r1 = 0 then
            return [x0, y0, r0]
        end if;
        q:= quo(r0, r1, z);
        r2:= expand(r0 - q*r1);
        x2:= expand(x0 - q*x1);
        y2:= expand(y0 - q*y1);
        r0:= r1;
        x0:= x1;
        y0:= y1;
        r1:= r2;
        x1:= x2;
        y1:= y2
    end do
end proc
```

```
> debug(polynomexgcd);
```

$$polynomexgcd \tag{8.2.75.2}$$

```
> polynomexgcd(z^3+z+1,z^2+2,z);
{--> enter polynomexgcd, args = z^3+z+1, z^2+2, z
```

$$x0 := 1$$

$$y0 := 0$$

$$r0 := z^3 + z + 1$$

$$x1 := 0$$

$$y1 := 1$$

$$r1 := z^2 + 2$$

$$q := z$$

$$r2 := 1 - z$$

$$x2 := 1$$

$$y2 := -z$$

$$r0 := z^2 + 2$$

$$x0 := 0$$

$$y0 := 1$$

$$r1 := 1 - z$$
$$x1 := 1$$
$$y1 := -z$$
$$q := -z - 1$$
$$r2 := 3$$
$$x2 := 1 + z$$
$$y2 := 1 - z^2 - z$$
$$r0 := 1 - z$$
$$x0 := 1$$
$$y0 := -z$$
$$r1 := 3$$
$$x1 := 1 + z$$
$$y1 := 1 - z^2 - z$$
$$q := \frac{1}{3} - \frac{1}{3} z$$
$$r2 := 0$$
$$x2 := \frac{2}{3} + \frac{1}{3} z^2$$
$$y2 := -\frac{1}{3} z - \frac{1}{3} - \frac{1}{3} z^3$$
$$r0 := 3$$
$$x0 := 1 + z$$
$$y0 := 1 - z^2 - z$$
$$r1 := 0$$
$$x1 := \frac{2}{3} + \frac{1}{3} z^2$$
$$y1 := -\frac{1}{3} z - \frac{1}{3} - \frac{1}{3} z^3$$

```
<-- exit polynomexgcd (now at top level) = [1+z, 1-z^2
-z, 3]}
```

$$[1 + z, 1 - z^2 - z, 3] \quad (8.2.75.3)$$

```
> undebug(polynomexgcd);
```

polynomexgcd (8.2.75.4)

```
> polynomexgcd(z^3+z+1,z^2+2,z);
```

$$[1 + z, 1 - z^2 - z, 3] \quad (8.2.75.5)$$

▶ ***8.2.76. Tétel.***

▶ ***8.2.77. Tétel.***

▶ **8.2.78. Tétel.*

▶ **8.2.79. Feladat.*

▶ **8.2.80. Maximális ideál.*

▶ **8.2.81. Következmény.*

▶ **8.2.82. Tétel.*

▶ **8.2.83. Prímideál.*

▶ **8.2.84. Tétel.*

▶ **8.2.85. Következmény.*

▶ **8.2.86. Tétel.*

▼ *8.2.87. Hányadostest.*

```
> r1:=(z^3-1)/(z^2-1); r1:=simplify(r1); r2:=simplify((z^4-1)
  /(z^3-1));
  r1*r2;
```

$$r1 := \frac{z^3 - 1}{z^2 - 1}$$

$$r1 := \frac{z^2 + z + 1}{1 + z}$$

$$r2 := \frac{z^3 + z^2 + z + 1}{z^2 + z + 1}$$

$$\frac{z^3 + z^2 + z + 1}{1 + z} \qquad (8.2.87.1)$$

▶ *8.2.88. Következmény.*

▶ **8.2.89. Algebrai struktúrák.*

▶ *->8.2.90. Feladat.*

▶ *->8.2.91. Feladat.*

▶ *->8.2.92. Feladat.*

▶ *->8.2.93. Feladat.*

▶ *->8.2.94. Feladat.*

▶ *8.2.95. Feladat.*

▶ *8.2.96. Feladat.*

▶ *8.2.97. Feladat.*

▶ *8.2.98. További feladatok.*

▼ 8.3. Polinomok

```
> restart;with(PolynomialTools);
```

[CoefficientList, CoefficientVector, GcdFreeBasis, GreatestFactorialFactorization, Hurwitz, IsSelfReciprocal, MinimalPolynomial, PDEToPolynomial, PolynomialToPDE, ShiftEquivalent, ShiftlessDecomposition, Shorten, Shorter, Sort, Split, Splits, Translate] (8.3.1)

8.3.1. Polinomok.

```
> p1:=2*x^2+x+3; p2:=5*x^3+9; p1+p2; p1*p2; expand(%);
```

$$p1 := 2\,x^2 + x + 3$$

$$p2 := 5\,x^3 + 9$$

$$2\,x^2 + x + 12 + 5\,x^3$$

$$\left(2\,x^2 + x + 3\right)\left(5\,x^3 + 9\right)$$

$$10\,x^5 + 18\,x^2 + 5\,x^4 + 9\,x + 15\,x^3 + 27 \qquad (8.3.1.1)$$

*8.3.2. Formális hatványsorok.

->8.3.3. Feladat.

->8.3.4. Feladat.

->8.3.5. Feladat.

->8.3.6. Feladat.

8.3.7. Polinomfüggvények.

```
> map(x->x mod 5,[0,1,2,3,4]); map(x->x^5 mod 5,[0,1,2,3,4]);
```

$$[0,\ 1,\ 2,\ 3,\ 4]$$

$$[0,\ 1,\ 2,\ 3,\ 4] \qquad (8.3.7.1)$$

8.3.8. A maradékos osztás tétele polinomokra.

```
> a:=5*x^4+9; b:=x^2+x+1; r:=rem(a,b,x); q:=quo(a,b,x);
  expand(q*b+r);
```

$$a := 5\,x^4 + 9$$

$$b := x^2 + x + 1$$

$$r := 9 + 5\,x$$

$$q := 5\,x^2 - 5\,x$$

$$5\,x^4 + 9 \qquad (8.3.8.1)$$

8.3.9. Következmény: gyöktényező leválasztása.

```
> quo(x^3-1,x-1,x,'r'); r;
```

$$x^2+x+1$$

$$0 \quad (8.3.9.1)$$

8.3.10. Következmény.

8.3.11. Következmény.

8.3.12. Következmény.

8.3.13. Következmény.

8.3.14. Megjegyzés.

```
> polydivisorx:=proc(a,b) local bb,db,r,dr,rr,rrr,q;
  if a=0 then return true fi;
  if b=0 then return false fi;
  bb:=lcoeff(b); db:=degree(b); r:=a;
  while degree(r)>=degree(b) do
    rr:=lcoeff(r); dr:=degree(r); rrr:=irem(rr,bb,'q');
    if rrr<>0 then return false fi;
    r:=expand(r-q*b*x^(dr-db));
  od; evalb(r=0); end;
```

(8.3.14.1)

```
polydivisorx := proc(a, b)
    local bb, db, r, dr, rr, rrr, q;
    if a = 0 then
        return true
    end if;
    if b = 0 then
        return false
    end if;
    bb := lcoeff(b);
    db := degree(b);
    r := a;
    while degree(b) <= degree(r) do
        rr := lcoeff(r);
        dr := degree(r);
        rrr := irem(rr, bb, 'q');
        if rrr <> 0 then
            return false
        end if;
```

$r := expand(r - q*b*x^\wedge(dr - db))$

end do;

$evalb(r = 0)$

end proc

```
> debug(polydivisorx);
```

polydivisorx (8.3.14.2)

```
> polydivisorx(4*x^2-1,2*x+1);
{--> enter polydivisorx, args = 4*x^2-1, 2*x+1
```

$$bb := 2$$
$$db := 1$$
$$r := 4x^2 - 1$$
$$rr := 4$$
$$dr := 2$$
$$rrr := 0$$
$$r := -1 - 2x$$
$$rr := -2$$
$$dr := 1$$
$$rrr := 0$$
$$r := 0$$
$$true$$

```
<-- exit polydivisorx (now at top level) = true}
```

true (8.3.14.3)

```
> undebug(polydivisorx);
```

polydivisorx (8.3.14.4)

```
> polydivisorx(4*x^2-1,2*x+1);
```

true (8.3.14.5)

*8.3.15. Megjegyzés: pszeudoosztás.

8.3.16. Megjegyzés: Horner-elrendezés.

```
> Horner:=proc(L::list,c) local i,LL,r; LL:=[];
  if nops(L)=0 then return LL,0 fi; r:=L[nops(L)];
  for i from nops(L)-1 to 1 by -1 do
    LL:=[r,op(LL)]; r:=L[i]+r*c;
  od; LL,r; end;
```

$Horner :=$ **proc**$(L::list, c)$ (8.3.16.1)

local i, LL, r;

$LL := [\,]$;

if $nops(L) = 0$ **then**

return $LL, 0$

end if;
$r := L[nops(L)]$;
for i **from** $nops(L) - 1$ **by** -1 **to** 1 **do**
$LL := [r, op(LL)]$;
$r := L[i] + r^* c$
end do;
LL, r
end proc

```
> p:=x^3+4*x^2-3*x+7; CoefficientList(p,x); Horner(%,2);
  quo(p,x-2,x); rem(p,x-2,x);
```

$$p := x^3 + 4 x^2 - 3 x + 7$$
$$[7, -3, 4, 1]$$
$$[9, 6, 1], 25$$
$$x^2 + 6 x + 9$$
$$25 \quad (8.3.16.2)$$

```
> convert(p,horner);
```

$$7 + (-3 + (4 + x) x) x \quad (8.3.16.3)$$

8.3.17. Megjegyzés.

```
> solve(x^2+1,x);
```

$$I, -I \quad (8.3.17.1)$$

```
> msolve(x^2+1,2);
```

$$\{x = 1\} \quad (8.3.17.2)$$

```
> msolve(x^2+1,3);
> msolve(x^2+1,5);
```

$$\{x = 2\}, \{x = 3\} \quad (8.3.17.3)$$

->8.3.18. Feladat.
->8.3.19. Feladat.
->8.3.20. Feladat.
->8.3.21. Feladat.
->8.3.22. Feladat.
->8.3.23. Feladat.
->8.3.24. Feladat.
->8.3.25. Feladat.
->8.3.26. Feladat.

*8.3.27. Körosztási polinomok.

*8.3.28. Wilson tétele.

```
> map(i->(i-1)!+1 mod i,[i$i=2..30]);
```

[0, 0, 3, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1] (8.3.28.1)

8.3.29. Polinom algebrai deriváltja.

```
> p:=x^4+3*x^2+2*x+1; diff(p,x);
```

$$p := x^4 + 3x^2 + 2x + 1$$

$$4x^3 + 6x + 2 \quad (8.3.29.1)$$

*8.3.30. Megjegyzés.

8.3.31. Tétel.

8.3.32. Következmény.

8.3.33. Többszörös gyökök.

8.3.34. Tétel.

8.3.35. Megjegyzés.

->8.3.36. Feladat.

->8.3.37. Feladat.

->8.3.38. Feladat.

->8.3.39. Feladat.

8.3.40. Irreducibilis polinomok és testbővítések.

```
> Split(p,x);
```

$$\Big(x + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big) + \mathrm{RootOf}\Big(\_Z^3 + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)\,\_Z^2 + \Big(3 + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)^2\Big)\,\_Z + 2 + 3\,\mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big) + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)^3\Big) + \mathrm{RootOf}\Big(\_Z^2 + \Big(\mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big) + \mathrm{RootOf}\Big(\_Z^3 + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)\,\_Z^2 + \Big(3 + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)^2\Big)\,\_Z + 2 + 3\,\mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big) + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)^3\Big)\Big)\,\_Z + 3 + \mathrm{RootOf}\big(\_Z^4 + 3\,\_Z^2 + 2\,\_Z + 1\big)^2 + \mathrm{RootOf}\Big(\_Z^3 + \mathrm{RootOf}\big(\_Z^4 \quad (8.3.40.1)$$

$$+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2+\big(3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)\,\mathrm{RootOf}\big(\_Z^4$$
$$+3\,\_Z^2+2\,\_Z+1\big)$$
$$+\mathrm{RootOf}\big(\_Z^3+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2+\big(3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)^2\big)\big)\,\big(x-\mathrm{RootOf}\big($$
$$\_Z^3+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2+\big(3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)\big)\,\big(x-\mathrm{RootOf}\big(\_Z^4$$
$$+3\,\_Z^2+2\,\_Z+1\big)\big)\,\big(x-\mathrm{RootOf}\big(\_Z^2+\big(\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^3+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2$$
$$+\big(3+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4$$
$$+3\,\_Z^2+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)\big)\,\_Z+3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2+\mathrm{RootOf}\big(\_Z^3+\mathrm{RootOf}\big(\_Z^4$$
$$+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2+\big(3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)\,\mathrm{RootOf}\big(\_Z^4$$
$$+3\,\_Z^2+2\,\_Z+1\big)$$
$$+\mathrm{RootOf}\big(\_Z^3+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)\,\_Z^2+\big(3$$
$$+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^2\big)\,\_Z+2+3\,\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2$$
$$+2\,\_Z+1\big)+\mathrm{RootOf}\big(\_Z^4+3\,\_Z^2+2\,\_Z+1\big)^3\big)^2\big)\big)$$

```
> p:=x^4; Nextpoly(p,x) mod 2; Nextpoly(%,x) mod 2; Prevpoly
  (%,x) mod 2;
```

$$p:=x^4$$
$$x^4+1$$
$$x^4+x$$
$$x^4+1 \qquad (8.3.40.2)$$

```
> p:=x^4; Nextprime(p,x) mod 2; Nextprime(%,x) mod 2;
  Prevprime(%,x) mod 2;
```

$$p:=x^4$$
$$x^4+x+1$$
$$x^4+x^3+1$$

$$x^4+x+1 \quad (8.3.40.3)$$

8.3.41. Példák.

```
> Split(x^2+1,x);
```

$$(x-\mathrm{RootOf}(\_Z^2+1))\,(x+\mathrm{RootOf}(\_Z^2+1)) \quad (8.3.41.1)$$

```
> p:=x^2+x+1; modpol(x^8+4*x^2,p,x,2); modpol(1/%,p,x,2);
  modpol(%*%%,p,x,2);
```

$$p:=x^2+x+1$$
$$x+1$$
$$x$$
$$1 \quad (8.3.41.2)$$

->8.3.42. Feladat.
->8.3.43. Feladat.
->8.3.44. Feladat.
->8.3.45. Feladat.
8.3.46. Feladat.
->8.3.47. Feladat.
8.3.48. Feladat.
8.3.49. Feladat.
8.3.50. Feladat.
8.3.51. Feladat.
8.3.52. Feladat.
8.3.53. Feladat.
8.3.54. Véges testek elemszáma.
8.3.55. Megjegyzések.
8.3.56. Feladat.
8.3.57. Alkalmazás: a Rijndael és AES blokkrejtjelzők.

```
> lgn:=8; n:=2^lgn-1; RijndaelPoly:=Nextprime(Z^lgn,Z) mod 2;
  alpha:=Z;
```

$$lgn:=8$$
$$n:=255$$
$$RijndaelPoly:=Z^8+Z^4+Z^3+Z+1$$
$$\alpha:=Z \quad (8.3.57.1)$$

```
> C:=Matrix([[1,0,0,0,1,1,1,1],[1,1,0,0,0,1,1,1],[1,1,1,0,0,
```

```
0,1,1],[1,1,1,1,0,0,0,1],[1,1,1,1,1,0,0,0],[0,1,1,1,1,1,0,
0],[0,0,1,1,1,1,1,0],[0,0,0,1,1,1,1,1]]);c:=Vector([1,1,0,
0,0,1,1,0]);
```

$$C := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$c := \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (8.3.57.2)$$

```
> with(LinearAlgebra);
```

[*&x*, *Add*, *Adjoint*, *BackwardSubstitute*, *BandMatrix*, *Basis*, *BezoutMatrix*, *BidiagonalForm*, *BilinearForm*, *CharacteristicMatrix*, *CharacteristicPolynomial*, *Column*, *ColumnDimension*, *ColumnOperation*, *ColumnSpace*, *CompanionMatrix*, *ConditionNumber*, *ConstantMatrix*, *ConstantVector*, *Copy*, *CreatePermutation*, *CrossProduct*, *DeleteColumn*, *DeleteRow*, *Determinant*, *Diagonal*, *DiagonalMatrix*, *Dimension*, *Dimensions*, *DotProduct*, *EigenConditionNumbers*, *Eigenvalues*, *Eigenvectors*, *Equal*, *ForwardSubstitute*, *FrobeniusForm*, *GaussianElimination*, *GenerateEquations*, *GenerateMatrix*, *GetResultDataType*, *GetResultShape*, *GivensRotationMatrix*, *GramSchmidt*, *HankelMatrix*, *HermiteForm*, *HermitianTranspose*, *HessenbergForm*, *HilbertMatrix*, *HouseholderMatrix*, (8.3.57.3)

IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, LA_Main, LUDecomposition, LeastSquares, LinearSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SubMatrix, SubVector, SumBasis, SylvesterMatrix, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

```
> Cinv:=MatrixInverse(C) mod 2;
```

$$Cinv := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad (8.3.57.4)$$

```
> S:=proc(x) local i,xx; global RijndaelPoly,C,c;
    xx:=convert(x,base,2);
    xx:=add(xx[i]*Z^(i-1),i=1..nops(xx));
    if xx<>0 then
      xx:=modpol(1/xx,RijndaelPoly,Z,2)
    else
      xx:=modpol(xx,RijndaelPoly,Z,2)
    fi;
    xx:=CoefficientList(xx,Z);
    while nops(xx)<8 do xx:=[op(xx),0] od;
    xx:=convert(xx,Vector);
    xx:=Multiply(C,xx) mod 2;
```

```
    xx:=Add(xx,c) mod 2;
    add(xx[i]*2^(i-1),i=1..8);
  end;

  S(0); S(1); S(2);
```

$S := \mathbf{proc}(x)$
  local i, xx;
  global $RijndaelPoly$, C, c;
  $xx := convert(x, base, 2)$;
  $xx := add(xx[i] * Z \wedge (i-1),\ i = 1 .. nops(xx))$;
  if $xx <> 0$ **then**
    $xx := modpol(1 / xx, RijndaelPoly, Z, 2)$
  else
    $xx := modpol(xx, RijndaelPoly, Z, 2)$
  end if;
  $xx := PolynomialTools:-CoefficientList(xx, Z)$;
  while $nops(xx) < 8$ **do**
    $xx := [op(xx), 0]$
  end do;
  $xx := convert(xx, Vector)$;
  $xx := mod(LinearAlgebra:-Multiply(C, xx), 2)$;
  $xx := mod(LinearAlgebra:-Add(xx, c), 2)$;
  $add(xx[i] * 2 \wedge (i-1),\ i = 1 .. 8)$
end proc

$$99$$
$$124$$
$$119 \qquad (8.3.57.5)$$

```
> Sinv:=proc(x) local i,xx; global RijndaelPoly,Cinv,c;
    xx:=convert(x,base,2);
    while nops(xx)<8 do xx:=[op(xx),0] od;
    xx:=convert(xx,Vector);
    xx:=Add(xx,c,1,-1) mod 2;
    xx:=Multiply(Cinv,xx) mod 2;
    xx:=add(xx[i]*Z^(i-1),i=1..8);
    if xx<>0 then xx:=modpol(1/xx,RijndaelPoly,Z,2) fi;
    xx:=CoefficientList(xx,Z);
    add(xx[i]*2^(i-1),i=1..nops(xx));
  end;

  Sinv(99); Sinv(124); Sinv(119);
```

```
Sinv := proc(x)
    local i, xx;
    global RijndaelPoly, Cinv, c;
    xx := convert(x, base, 2);
    while nops(xx) < 8 do
        xx := [op(xx),
        0]
    end do;
    xx := convert(xx, Vector);
    xx := mod(LinearAlgebra:-Add(xx, c, 1, -1), 2);
    xx := mod(LinearAlgebra:-Multiply(Cinv, xx), 2);
    xx := add(xx[i]*Z^(i - 1), i = 1 ..8);
    if xx <> 0 then
        xx := modpol(1 / xx, RijndaelPoly, Z, 2)
    end if;
    xx := PolynomialTools:-CoefficientList(xx, Z);
    add(xx[i]*2^(i - 1), i = 1 ..nops(xx))
end proc
```

0

1

2 (8.3.57.6)

```
> X:=proc(x,y,b) local u,v,i,xx,yy; # bitwise xor in b bit
  length
    xx:=convert(x,base,2); yy:=convert(y,base,2);
    while nops(xx)<b do xx:=[op(xx),0] od;
    while nops(yy)<b do yy:=[op(yy),0] od;
    xx:=zip((u,v)->u+v mod 2,xx,yy);
    add(xx[i]*2^(i-1),i=1..b);
  end;

  X(5,3,4);
```

```
X := proc(x, y, b)
    local u, v, i, xx, yy;
    xx := convert(x, base, 2);
    yy := convert(y, base, 2);
    while nops(xx) < b do
        xx := [op(xx),
        0]
```

end do;
while *nops*(*yy*) < *b* **do**
yy := [*op*(*yy*), 0]
end do;
xx := *zip*(**proc**(*u*, *v*)
option *operator*, *arrow*;
mod(*u* + *v*, 2)
end proc, *xx*, *yy*);
add(*xx*[*i*]*2^(*i* − 1), *i* = 1 ..*b*)
end proc

$$6 \tag{8.3.57.7}$$

```
> Word2Bytes:=proc(x) local xx;
    xx:=convert(x,base,256);
    while nops(xx)<4 do xx:=[op(xx),0] od;
    [xx[4],xx[3],xx[2],xx[1]];
  end;

  Bytes2Word:=proc(x) local i; add(x[i]*256^(4-i),i=1..4)
  end;

  Bytes2Word([0,1,2,3]); Word2Bytes(%);
```

Word2Bytes := **proc**(*x*)
local *xx*;
xx := *convert*(*x*, *base*, 256);
while *nops*(*xx*) < 4 **do**
xx := [*op*(*xx*), 0]
end do;
[*xx*[4], *xx*[3],
xx[2], *xx*[1]]
end proc
Bytes2Word := **proc**(*x*)
local *i*;
add(*x*[*i*]*256^(4 − *i*), *i* = 1 ..4)
end proc

$$66051$$
$$[0, 1, 2, 3] \tag{8.3.57.8}$$

```
> K:=["00010203","05060708","0A0B0C0D","0F101112"];K:=map(x-
  >convert(x,decimal,hex),K);
```

K := ["00010203", "05060708", "0A0B0C0D", "0F101112"]
(8.3.57.9)

$$K := [66051, 84281096, 168496141, 252711186] \quad (8.3.57.9)$$

```
> RijndaelKeys:=proc(K,kk) local i,t,x,tt,k,KK,R; global
  RijndaelPoly;
    k:=nops(K); KK:=K;
    R:=1;
    for i from k to kk-1 do
      t:=KK[i];
      if k>6 and (i mod k=4) then
        t:=Word2Bytes(t);
        t:=map(x->S(x),t);
        t:=Bytes2Word(t);
      fi;
      if i mod k=0 then
        t:=Word2Bytes(t);
        t:=[t[2],t[3],t[4],t[1]];
        t:=map(x->S(x),t);
        tt:=CoefficientList(R,Z);
        tt:=add(tt[i]*2^(i-1),i=1..nops(tt));
        tt:=X(t[1],tt,8);
        t:=[tt,t[2],t[3],t[4]];
        R:=modpol(R*Z,RijndaelPoly,Z,2);
        t:=Bytes2Word(t);
      fi;
      t:=X(t,KK[i-k+1],32);
      KK:=[op(KK),t];
    od;
    KK;
  end;

  RijndaelKeys(K,8);
```

RijndaelKeys := **proc**(*K*, *kk*)
    local *i*, *t*, *x*, *tt*, *k*, *KK*, *R*;
    global *RijndaelPoly*;
    $k := nops(K)$;
    $KK := K$;
    $R := 1$;
    for *i* **from** *k* **to** $kk - 1$ **do**
        $t := KK[i]$;
        if $6 < k$ **and** $mod(i,$
        $k) = 4$ **then**
            $t := Word2Bytes(t)$;
            $t := map($**proc**(x)
                option *operator*, *arrow*;

```
                S(x)
            end proc, t);
            t := Bytes2Word(t)
        end if;
        if mod(i, k) = 0 then
            t := Word2Bytes(t);
            t := [t[2], t[3], t[4], t[1]];
            t := map(proc(x)
                option operator, arrow;
                S(x)
            end proc, t);
            tt := PolynomialTools:-CoefficientList(R, Z);
            tt := add(tt[i]*2^(i - 1), i = 1 ..nops(tt));
            tt := X(t[1],
            tt, 8);
            t := [tt, t[2], t[3], t[4]];
            R := modpol(R*Z,
            RijndaelPoly, Z, 2);
            t := Bytes2Word(t)
        end if;
        t := X(t,
        KK[i - k + 1], 32);
        KK := [op(KK), t]
    end do;
    KK
end proc
```

[66051, 84281096, 168496141, 252711186, 3414412149,
3464875133, 3297689712, 3416183138] (8.3.57.10)

```
> SubBytes:=proc(L) local x,y; global S;
    map(x->map(y->S(y),x),L);
  end;
```

SubBytes := proc(L) (8.3.57.11)

```
    local x, y;
    global S;
    map(proc(x)
        option operator, arrow;
        map(proc(y)
```

option *operator*, *arrow*;
S(*y*)
end proc, *x*)
end proc, *L*)
end proc

```
> SubBytesinv:=proc(L) local x,y; global Sinv;
    map(x->map(y->Sinv(y),x),L);
  end;
```

SubBytesinv := **proc**(*L*) (8.3.57.12)
local *x*, *y*;
global *Sinv*;
map(**proc**(*x*)
option *operator*, *arrow*;
map(**proc**(*y*)
option *operator*, *arrow*;
Sinv(*y*)
end proc, *x*)
end proc, *L*)
end proc

```
> ShiftRow:=proc(L) local b,i,d,LL,LLL; b:=nops(L); LLL:=[];
    if b>6 then d:=1 else d:=0 fi;
    for i to b do
      LL:=[L[i][1],L[1+(i mod b)][2],L[1+(i+1+d mod b)][3],L
  [1+(i+2+d mod b)][4]];
      LLL:=[op(LLL),LL];
    od; LLL;
  end;

  ShiftRow([[0,1,2,3],[10,11,12,13],[20,21,22,23],[30,31,32,
  33]]);
```

ShiftRow := **proc**(*L*)
local *b*, *i*, *d*, *LL*, *LLL*;
b := *nops*(*L*);
LLL := [ ];
if 6 < *b* **then**
d := 1
else
d := 0
end if;

```
    for i to b do
        LL := [L[i][1], L[1 + (mod(i, b))][2],
        L[1 + (mod(i+1 + d, b))][3],
        L[1 + (mod(i+2 + d, b))][4]];
        LLL := [op(LLL), LL]
    end do;
    LLL
end proc
```

[[0, 11, 22, 33], [10, 21, 32, 3], [20, 31, 2, 13], [30, 1, 12, 23]] (8.3.57.13)

```
> ShiftRowinv:=proc(L) local b,i,d,LL,LLL; b:=nops(L); LLL:=
[];
  if b>6 then d:=1 else d:=0 fi;
  for i to b do
    LL:=[L[i][1],L[1+(i-2 mod b)][2],L[1+(i-3-d mod b)][3],
L[1+(i-4-d mod b)][4]];
    LLL:=[op(LLL),LL];
  od; LLL;
end;

ShiftRowinv([[0,1,2,3],[10,11,12,13],[20,21,22,23],[30,31,
32,33]]);
```

```
ShiftRowinv := proc(L)
    local b, i, d, LL, LLL;
    b := nops(L);
    LLL := [];
    if 6 < b then
        d := 1
    else
        d := 0
    end if;
    for i to b do
        LL := [L[i][1], L[1 + (mod(i - 2, b))][2],
        L[1 + (mod(i - 3 - d, b))][3],
        L[1 + (mod(i - 4 - d, b))][4]];
        LLL := [op(LLL), LL]
    end do;
    LLL
end proc
```

[[0, 31, 22, 13], [10, 1, 32, 23], [20, 11, 2, 33], [30, 21, 12, 3]] (8.3.57.14)

```
> normalizepolyzZ:=proc(p)
  local i; global RijndaelPoly;
  CoefficientList(expand(p) mod 2,z);
  map(x->modpol(x,RijndaelPoly,Z,2),%);
  add(%[i]*z^(i-1),i=1..nops(%)); sort(%); end;
```

normalizepolyzZ := **proc**(*p*) (8.3.57.15)

local *i*;
global *RijndaelPoly*;
PolynomialTools:-*CoefficientList*(*mod*(*expand*(*p*), 2), *z*);
map(**proc**(*x*)
option *operator*, *arrow*;
modpol(*x*, *RijndaelPoly*, *Z*, 2)
end proc, \`%\`);
add(\`%\`[*i*]*z^(*i* − 1), *i* = 1 ..*nops*(\`%\`));
sort(\`%\`)
end proc

```
> RijndaelMixPoly:=(Z+1)*z^3+z^2+z+Z;

  MixMul:=proc(L) local p,LL,x,i;
    global RijndaelPoly,RijndaelMixPoly;
    LL:=map(x->convert(x,base,2),L);
    LL:=map(x->add(x[i]*Z^(i-1),i=1..nops(x)),LL);
    p:=add(LL[i]*z^(i-1),i=1..4);
    p:=p*RijndaelMixPoly;
    p:=normalizepolyzZ(p);
    while degree(p,z)>=4 do
      p:=normalizepolyzZ(p-lcoeff(p,z)*z^(degree(p,z)-4)*
  (z^4+1) mod 2);
    od;
    LL:=CoefficientList(p,z);
    while nops(LL)<4 do LL:=[op(LL),0] od;
    map(x->subs(Z=2,x),LL);
  end;

  MixMul([1,1,1,1]); MixMul([219,19,83,69]); MixMul([242,10,
  34,92]);
  MixMul([198,198,198,198]); MixMul([212,212,212,213]);
  MixMul([45,38,49,76]);
```

$$RijndaelMixPoly := (Z+1)\,z^3 + z^2 + z + Z$$

MixMul := **proc**(*L*)
local *p*, *LL*, *x*, *i*;

```
    global RijndaelPoly,
   RijndaelMixPoly;
    LL := map(proc(x)
        option operator, arrow;
        convert(x, base, 2)
    end proc, L);
    LL := map(proc(x)
        option operator, arrow;
        add(x[i]*Z^(i - 1), i = 1..nops(x))
    end proc, LL);
    p := add(LL[i]*z^(i - 1), i = 1..4);
    p := p*RijndaelMixPoly;
    p := normalizepolyzZ(p);
    while 4 <= degree(p, z) do
        p := normalizepolyzZ(mod(p - lcoeff(p, z)*z^(degree(p,
        z) - 4)*(z^4 + 1), 2))
    end do;
    LL := PolynomialTools:-CoefficientList(p, z);
    while nops(LL) < 4 do
        LL := [op(LL), 0]
    end do;
    map(proc(x)
        option operator, arrow;
        subs(Z = 2, x)
    end proc, LL)
end proc
```

[1, 1, 1, 1]

[142, 77, 161, 188]

[159, 220, 88, 157]

[198, 198, 198, 198]

[213, 213, 215, 214]

[77, 126, 189, 248] (8.3.57.16)

```
> RijndaelMixPolyinv:=(Z^3+Z+1)*z^3+(Z^3+Z^2+1)*z^2+(Z^3+1)*
  z+Z^3+Z^2+Z;

  MixMulinv:=proc(L) local p,LL,x,i;
    global RijndaelPoly,RijndaelMixPolyinv;
    LL:=map(x->convert(x,base,2),L);
    LL:=map(x->add(x[i]*Z^(i-1),i=1..nops(x)),LL);
```

```
    p:=add(LL[i]*z^(i-1),i=1..4);
    p:=p*RijndaelMixPolyinv;
    p:=normalizepolyzZ(p);
    while degree(p,z)>=4 do
      p:=normalizepolyzZ(p-lcoeff(p,z)*z^(degree(p,z)-4)*
  (z^4+1) mod 2);
    od;
    LL:=CoefficientList(p,z);
    while nops(LL)<4 do LL:=[op(LL),0] od;
    map(x->subs(Z=2,x),LL);
  end;

  MixMulinv([1,1,1,1]); MixMulinv([142,77,161,188]);
  MixMulinv([159,220,88,157]); MixMulinv([198,198,198,198]);
  MixMulinv([213,213,215,214]); MixMulinv([77,126,189,248]);
```

$$RijndaelMixPolyinv := (Z^3+Z+1)\,z^3 + (Z^3+Z^2+1)\,z^2 + (Z^3+1)\,z + Z^3 + Z^2 + Z$$

MixMulinv := **proc**(*L*)
    local *p*, *LL*, *x*, *i*;
    global *RijndaelPoly*,
    RijndaelMixPolyinv;
    LL := *map*(**proc**(*x*)
        option *operator*,
        arrow;
        convert(*x*, *base*, 2)
    end proc, *L*);
    LL := *map*(**proc**(*x*)
        option *operator*, *arrow*;
        add(*x*[*i*] * *Z*^(*i* − 1), *i* = 1 ..*nops*(*x*))
    end proc, *LL*);
    p := *add*(*LL*[*i*] * *z*^(*i* − 1), *i* = 1 ..4);
    p := *p* * *RijndaelMixPolyinv*;
    p := *normalizepolyzZ*(*p*);
    while 4 <= *degree*(*p*, *z*) **do**
        p := *normalizepolyzZ*(*mod*(*p* − *lcoeff*(*p*, *z*) * *z*^(*degree*(*p*,
        z) − 4) * (*z*^4 + 1), 2))
    end do;
    LL := *PolynomialTools*:-*CoefficientList*(*p*, *z*);
    while *nops*(*LL*) < 4 **do**
        LL := [*op*(*LL*), 0]

```
    end do;
    map(proc(x)
        option operator, arrow;
        subs(Z = 2, x)
    end proc, LL)
end proc
```

$$[1, 1, 1, 1]$$
$$[219, 19, 83, 69]$$
$$[242, 10, 34, 92]$$
$$[198, 198, 198, 198]$$
$$[212, 212, 212, 213]$$
$$[45, 38, 49, 76] \quad (8.3.57.17)$$

```
> Rijndael:=proc(M::list(posint),K::list(posint),r::posint)
    local i,j,k,m,KK,MM,MMM; k:=nops(K); m:=nops(M); MMM:=[];
    KK:=RijndaelKeys(K,m*(r+1));
    for j to m do
      MMM:=[op(MMM),X(KK[j],M[j],32)];
    od;
    for i from 2 to r do
      MM:=map(x->Word2Bytes(x),MMM);
      MM:=SubBytes(MM);
      MM:=ShiftRow(MM);
      MM:=map(x->MixMul(x),MM);
      MM:=map(x->Bytes2Word(x),MM);
      MMM:=[];
      for j to m do
        MMM:=[op(MMM),X(KK[(i-1)*m+j],MM[j],32)];
      od;
    od;
    MM:=map(x->Word2Bytes(x),MMM);
    MM:=SubBytes(MM);
    MM:=ShiftRow(MM);
    MM:=map(x->Bytes2Word(x),MM);
    MMM:=[];
    for j to m do
      MMM:=[op(MMM),X(KK[r*m+j],MM[j],32)];
    od;
    MMM;
  end;

  M:=["506812A4","5F08C889","B97F5980","038B8359"];
  M:=map(x->convert(x,decimal,hex),M);
  Rijndael(M,K,10);
```

Rijndael := proc(M::(list(posint)), K::(list(posint)), r::posint)

```
local i, j, k, m, KK, MM, MMM;
k := nops(K);
m := nops(M);
MMM := [ ];
KK := RijndaelKeys(K, m*(r+1));
for j to m do
    MMM := [op(MMM), X(KK[j], M[j], 32)]
end do;
for i from 2 to r do
    MM := map(proc(x)
        option operator, arrow;
        Word2Bytes(x)
    end proc, MMM);
    MM := SubBytes(MM);
    MM := ShiftRow(MM);
    MM := map(proc(x)
        option operator, arrow;
        MixMul(x)
    end proc, MM);
    MM := map(proc(x)
        option operator,
       arrow;
        Bytes2Word(x)
    end proc, MM);
    MMM := [ ];
    for j to m do
        MMM := [op(MMM), X(KK[(i-1)*m+j],
       MM[j], 32)]
    end do
end do;
MM := map(proc(x)
    option operator, arrow;
    Word2Bytes(x)
end proc, MMM);
MM := SubBytes(MM);
MM := ShiftRow(MM);
MM := map(proc(x)
    option operator, arrow;
```

```
        Bytes2Word(x)
    end proc, MM);
    MMM := [ ];
    for j to m do
        MMM := [op(MMM), X(KK[r* m + j], MM[j], 32)]
    end do;
    MMM
end proc
```

$M := ["506812A4", "5F08C889", "B97F5980", "038B8359"]$

$M := [1348997796, 1594411145, 3112130944, 59474777]$

$$[3639947859, 2190077821, 112527012, 4250659273] \quad (8.3.57.18)$$

```
> Rijndaelinv:=proc(M::list(posint),K::list(posint),
  r::posint)
    local i,j,k,m,KK,MM,MMM; k:=nops(K); m:=nops(M); MMM:=[];
    KK:=RijndaelKeys(K,m*(r+1));
    for j to m do
      MMM:=[op(MMM),X(KK[r*m+j],M[j],32)];
    od;
    MM:=map(x->Word2Bytes(x),MMM);
    MM:=ShiftRowinv(MM);
    MM:=SubBytesinv(MM);
    MM:=map(x->Bytes2Word(x),MM);
    for i from r to 2 by -1 do
      MMM:=[];
      for j to m do
        MMM:=[op(MMM),X(KK[(i-1)*m+j],MM[j],32)];
      od;
      MM:=map(x->Word2Bytes(x),MMM);
      MM:=map(x->MixMulinv(x),MM);
      MM:=ShiftRowinv(MM);
      MM:=SubBytesinv(MM);
      MM:=map(x->Bytes2Word(x),MM);
    od;
    MMM:=[];
    for j to m do
      MMM:=[op(MMM),X(KK[j],MM[j],32)];
    od;
    MMM;
  end;

  CM:=["D8F53253","8289EF7D","06B506A4","FD5BE9C9"];
  CM:=map(x->convert(x,decimal,hex),CM);
  Rijndaelinv(CM,K,10);
  M:=["506812A4","5F08C889","B97F5980","038B8359"];
  M:=map(x->convert(x,decimal,hex),M);
```

```
Rijndaelinv := proc(M::(list(posint)), K::(list(posint)), r::posint)
    local i, j, k, m, KK, MM, MMM;
    k := nops(K);
    m := nops(M);
    MMM := [ ];
    KK := RijndaelKeys(K, m*(r+1));
    for j to m do
        MMM := [op(MMM), X(KK[r*m+j], M[j], 32)]
    end do;
    MM := map(proc(x)
        option operator, arrow;
        Word2Bytes(x)
    end proc, MMM);
    MM := ShiftRowinv(MM);
    MM := SubBytesinv(MM);
    MM := map(proc(x)
        option operator, arrow;
        Bytes2Word(x)
    end proc, MM);
    for i from r by -1 to 2 do
        MMM := [ ];
        for j to m do
            MMM := [op(MMM), X(KK[(i-1)*m+j], MM[j], 32)]
        end do;
        MM := map(proc(x)
            option operator, arrow;
            Word2Bytes(x)
        end proc, MMM);
        MM := map(proc(x)
            option operator, arrow;
            MixMulinv(x)
        end proc, MM);
        MM := ShiftRowinv(MM);
        MM := SubBytesinv(MM);
        MM := map(proc(x)
            option operator, arrow;
            Bytes2Word(x)
```

```
            end proc, MM)
        end do;
        MMM := [ ];
        for j to m do
            MMM := [op(MMM), X(KK[j], MM[j], 32)]
        end do;
        MMM
    end proc
```

$CM := ["D8F53253", "8289EF7D", "06B506A4", "FD5BE9C9"]$

$CM := [3639947859, 2190077821, 112527012, 4250659273]$

$[1348997796, 1594411145, 3112130944, 59474777]$

$M := ["506812A4", "5F08C889", "B97F5980", "038B8359"]$

$$M := [1348997796, 1594411145, 3112130944, 59474777] \quad (8.3.57.19)$$

8.3.58. Tétel.

****8.3.59. Tétel.***

8.3.60. Következmény.

8.3.61. Következmény.

****8.3.62. Feladat.***

****8.3.63. Feladat.***

8.3.64. Irreducibilis polinomok.

```
> factor(x^3-1); factor(6*x^2+12*x+12);
```

$$(x-1)\left(x^2+x+1\right)$$

$$6\,x^2+12\,x+12 \quad (8.3.64.1)$$

****8.3.65. Primitív polinomok.***

****8.3.66. Gauss lemmája.***

****8.3.67. Lemma.***

8.3.68. Gauss tétele.

****8.3.69. Legnagyobb közös osztó számolása Gauss-gyűrű feletti polinomgyűrű esetén.***

->8.3.70. Feladat.

->8.3.71. Feladat.

->8.3.72. Feladat.

->8.3.73. Feladat.

->8.3.74. Feladat.
->8.3.75. Feladat.
8.3.76. Feladat.
*8.3.77. Schönemann-Eisenstein-tétel.
*8.3.78. Következmény.

```
> factor(x^4+7);
```

$$x^4 + 7 \quad (8.3.78.1)$$

```
> (x^5-1)/(x-1); simplify(%); Translate(%,x,1);
```

$$\frac{x^5 - 1}{x - 1}$$

$$x^4 + x^3 + x^2 + x + 1$$

$$5 + 10\,x + 10\,x^2 + 5\,x^3 + x^4 \quad (8.3.78.2)$$

*8.3.79. Megjegyzések.
->8.3.80. Feladat.
8.3.81. Feladat.
8.3.82. Lagrange-interpoláció.

```
> with(CurveFitting);
```

[*BSpline*, *BSplineCurve*, *Interactive*, *LeastSquares*, *PolynomialInterpolation*, *RationalInterpolation*, *Spline*, *ThieleInterpolation*] (8.3.82.1)

```
> PolynomialInterpolation([[0,2],[1,4],[3,7],[4,5]],x);
```

$$-\frac{1}{4}\,x^3 + \frac{5}{6}\,x^2 + \frac{17}{12}\,x + 2 \quad (8.3.82.2)$$

```
> PolynomialInterpolation([[0,2],[1,4],[3,7],[4,5]],x,form=
  Lagrange);
```

$$-\frac{1}{6}\,(x-1)\,(x-3)\,(x-4) + \frac{2}{3}\,x\,(x-3)\,(x-4) - \frac{7}{6}\,x\,(x-1)\,(x-4) + \frac{5}{12}\,x\,(x-1)\,(x-3) \quad (8.3.82.3)$$

8.3.83. Titokmegosztás.

```
> p:=nextprime(10^50);
  t:=12345678901234567890123456789012345678901234567890;
```

```
P:=t+94536789321234890676867542309176345623498734562349*
x+79888998766657439865923165346578934524359870127935*x^2
mod p;
```

$p :=$
$1000000000000000000000000000000000000000000000000\backslash$
151

$t := 1234567890123456789012345678901234567890123456789 0$

$P := 12345678901234567890123456789012345678901234567890 + 94536789321234890676867542309176345623498734562349\, x + 79888998766657439865923165346578934524359870127935\, x^2$ (8.3.83.1)

```
> y1:=subs(x=1,P) mod p;
  y2:=subs(x=2,P) mod p;
  y3:=subs(x=3,P) mod p;
  y4:=subs(x=4,P) mod p;
  y5:=subs(x=5,P) mod p;
```

$y1 := 86771466989126898432914164444767625826759839258023$
$y2 := 20975252610334108707551202793680775023338184203573$
$y3 := 14957035764856198714034571835751793268636269404842$
$y4 := 68716816452693168452364271570980680562654094861830$
$y5 := 82254594673845017922540301999367436905391660574386$ (8.3.83.2)

```
> Interp([1,3,5],[y1,y3,y5],x) mod p;
```

$12345678901234567890123456789012345678901234567890 + 94536789321234890676867542309176345623498734562349\, x + 79888998766657439865923165346578934524359870127935\, x^2$ (8.3.83.3)

->8.3.84. Feladat.

8.3.85. Kronecker-eljárás.

```
> with(combinat);
```

[Chi, *bell*, binomial, *cartprod*, *character*, *choose*, *composition*, *conjpart*, *decodepart*, *encodepart*, *fibonacci*, *firstpart*, *graycode*, *inttovec*, *lastpart*, *multinomial*, *nextpart*, *numbcomb*, (8.3.85.1)

numbcomp, *numbpart*, *numbperm*, *partition*, *permute*,
powerset, *prevpart*, *randcomb*, *randpart*, *randperm*,
setpartition, stirling1, stirling2, *subsets*, *vectoint*]

```
> with(numtheory);
```

[*GIgcd*, *bigomega*, *cfrac*, *cfracpol*, *cyclotomic*, *divisors*, *factorEQ*, *factorset*, *fermat*, *imagunit*, *index*, *integral_basis*, *invcfrac*, *invphi*, *issqrfree*, *jacobi*, *kronecker*, λ, *legendre*, *mcombine*, *mersenne*, *migcdex*, *minkowski*, *mipolys*, *mlog*, *mobius*, *mroot*, *msqrt*, *nearestp*, *nthconver*, *nthdenom*, *nthnumer*, *nthpow*, *order*, *pdexpand*, ϕ, π, *pprimroot*, *primroot*, *quadres*, *rootsunity*, *safeprime*, σ, *sq2factor*, *sum2sqr*, τ, *thue*] (8.3.85.2)

```
> Kroneckerfactorx:=proc(p) local d,D,pp,i,X,Y,v,V,c,q,y;
  pp:=expand(p);
  if pp=0 then return [0] fi;
  if pp=1 then return [] fi;
  if pp=-1 then return [-1] fi;
  D:=degree(pp); X:=[]; Y:=[]; i:=0;
  while nops(X)<=floor(D/2) do
    v:=subs(x=i,pp);
    if v=0 then return [x-i,op(Kroneckerfactorx(quo(pp,x-i,x)
  ))] fi;
    V:=divisors(v); V:=V union map(y->-y,V);
    V:=[op(V)]; V:=sort(V,(u,v)->abs(u)<abs(v));
    X:=[op(X),i]; Y:=[op(Y),V];
    if i<=0 then i:=1-i else i:=-i fi;
  od;
  for d from 0 while D>=2*d do
    y:=cartprod(Y[1..d+1]);
    while not y[finished] do
      q:=PolynomialInterpolation(X[1..d+1],y[nextvalue](),x);
      if q=1 or q=-1 then next fi;
      if polydivisorx(pp,q) then
        return [q,op(Kroneckerfactorx(quo(pp,q,x)))] fi;
    od;
  od; [pp]; end;
```

Kroneckerfactorx := **proc**(*p*) (8.3.85.3)
 local *d*, D, *pp*, *i*, *X*, *Y*, *v*, *V*, *c*, *q*, *y*;
 pp := *expand*(*p*);
 if *pp* = 0 **then**
  return [0]
 end if;
 if *pp* = 1 **then**

```
            return [ ]
        end if;
        if pp = −1 then
            return [−1]
        end if;
        D := degree(pp);
        X:= [ ];
        Y:= [ ];
        i:= 0;
        while nops(X) <= floor(1 / 2 * D) do
            v:= subs(x = i, pp);
            if v = 0 then
                return [x − i, op(Kroneckerfactorx(quo(pp,
                x − i, x)))]
            end if;
            V:= numtheory:-divisors(v);
            V:= union(V, map(proc(y)
                option operator, arrow;
                −y
            end proc, V));
            V:= [op(V)];
            V:= sort(V, proc(u, v)
                option operator, arrow;
                abs(u) < abs(v)
            end proc);
            X:= [op(X), i];
            Y:= [op(Y), V];
            if i <= 0 then
                i:= 1 − i
            else
                i:= −i
            end if
        end do;
        for d from 0 while 2 * d <= D do
            y:= combinat:-cartprod(Y[1 ..d + 1]);
            while not y[finished] do
                q := CurveFitting:-PolynomialInterpolation(X[1 ..d + 1],
                y[nextvalue](), x);
```

```
            if q = 1 or q = -1 then
                next
            end if;
            if polydivisorx(pp, q) then
                return [q,
                op(Kroneckerfactorx(quo(pp, q, x)))]
            end if
        end do
    end do;
    [pp]
end proc
```

```
> debug(Kroneckerfactorx);
```

$$Kroneckerfactorx \tag{8.3.85.4}$$

```
> p:=x^5-2*x^4-2*x^3+4*x^2+x-2; Kroneckerfactorx(p);
```

$$p := x^5 - 2\,x^4 - 2\,x^3 + 4\,x^2 + x - 2$$

```
{--> enter Kroneckerfactorx, args = x^5-2*x^4-2*x^3+4*
x^2+x-2
```

$$pp := x^5 - 2\,x^4 - 2\,x^3 + 4\,x^2 + x - 2$$

$$\mathrm{D} := 5$$

$$X := [\,]$$

$$Y := [\,]$$

$$i := 0$$

$$v := -2$$

$$V := \{1, 2\}$$

$$V := \{-2, -1, 1, 2\}$$

$$V := [-2, -1, 1, 2]$$

$$V := [1, -1, 2, -2]$$

$$X := [0]$$

$$Y := [[1, -1, 2, -2]]$$

$$i := 1$$

$$v := 0$$

```
{--> enter Kroneckerfactorx, args = x^4-x^3-3*x^2+x+2
```

$$pp := x^4 - x^3 - 3\,x^2 + x + 2$$

$$\mathrm{D} := 4$$

$$X := [\,]$$

$$Y := [\,]$$

$$i := 0$$

$$v := 2$$

$$V := \{1, 2\}$$
$$V := \{-2, -1, 1, 2\}$$
$$V := [-2, -1, 1, 2]$$
$$V := [1, -1, 2, -2]$$
$$X := [0]$$
$$Y := [[1, -1, 2, -2]]$$
$$i := 1$$
$$v := 0$$

```
{--> enter Kroneckerfactorx, args = x^3-3*x-2
```

$$pp := x^3 - 3x - 2$$
$$\mathrm{D} := 3$$
$$X := []$$
$$Y := []$$
$$i := 0$$
$$v := -2$$
$$V := \{1, 2\}$$
$$V := \{-2, -1, 1, 2\}$$
$$V := [-2, -1, 1, 2]$$
$$V := [1, -1, 2, -2]$$
$$X := [0]$$
$$Y := [[1, -1, 2, -2]]$$
$$i := 1$$
$$v := -4$$
$$V := \{1, 2, 4\}$$
$$V := \{-4, -2, -1, 1, 2, 4\}$$
$$V := [-4, -2, -1, 1, 2, 4]$$
$$V := [1, -1, 2, -2, 4, -4]$$
$$X := [0, 1]$$
$$Y := [[1, -1, 2, -2], [1, -1, 2, -2, 4, -4]]$$
$$i := -1$$
$$y := table([finished = false, nextvalue = \mathbf{proc}()\ \ldots\ \mathbf{end\ proc}])$$
$$q := 1$$
$$q := -1$$
$$q := 2$$
$$q := -2$$
$$y := table([finished = false, nextvalue = \mathbf{proc}()\ \ldots\ \mathbf{end\ proc}])$$
$$q := 1$$

$$q := -2\,x + 1$$
$$q := x + 1$$
```
{--> enter Kroneckerfactorx, args = x^2-x-2
```
$$pp := x^2 - x - 2$$
$$\mathrm{D} := 2$$
$$X := [\,]$$
$$Y := [\,]$$
$$i := 0$$
$$v := -2$$
$$V := \{1, 2\}$$
$$V := \{-2, -1, 1, 2\}$$
$$V := [-2, -1, 1, 2]$$
$$V := [1, -1, 2, -2]$$
$$X := [0]$$
$$Y := [[1, -1, 2, -2]]$$
$$i := 1$$
$$v := -2$$
$$V := \{1, 2\}$$
$$V := \{-2, -1, 1, 2\}$$
$$V := [-2, -1, 1, 2]$$
$$V := [1, -1, 2, -2]$$
$$X := [0, 1]$$
$$Y := [[1, -1, 2, -2], [1, -1, 2, -2]]$$
$$i := -1$$
$$y := table([finished = false, nextvalue = \mathbf{proc}()\ \ldots\ \mathbf{end\ proc}])$$
$$q := 1$$
$$q := -1$$
$$q := 2$$
$$q := -2$$
$$y := table([finished = false, nextvalue = \mathbf{proc}()\ \ldots\ \mathbf{end\ proc}])$$
$$q := 1$$
$$q := -2\,x + 1$$
$$q := x + 1$$
```
{--> enter Kroneckerfactorx, args = x-2
```
$$pp := x - 2$$
$$\mathrm{D} := 1$$
$$X := [\,]$$

$$Y := [\,]$$
$$i := 0$$
$$v := -2$$
$$V := \{1, 2\}$$
$$V := \{-2, -1, 1, 2\}$$
$$V := [-2, -1, 1, 2]$$
$$V := [1, -1, 2, -2]$$
$$X := [0]$$
$$Y := [[1, -1, 2, -2]]$$
$$i := 1$$
$$y := table([finished = false, nextvalue = \mathbf{proc}()\ \ldots\ \mathbf{end\ proc}])$$
$$q := 1$$
$$q := -1$$
$$q := 2$$
$$q := -2$$
$$[x - 2]$$

```
<-- exit Kroneckerfactorx (now in Kroneckerfactorx) =
[x-2]}
<-- exit Kroneckerfactorx (now in Kroneckerfactorx) =
[x+1, x-2]}
<-- exit Kroneckerfactorx (now in Kroneckerfactorx) =
[x+1, x+1, x-2]}
<-- exit Kroneckerfactorx (now in Kroneckerfactorx) =
[x-1, x+1, x+1, x-2]}
<-- exit Kroneckerfactorx (now at top level) = [x-1, x
-1, x+1, x+1, x-2]}
```

$$[x - 1, x - 1, x + 1, x + 1, x - 2] \quad (8.3.85.5)$$

```
> undebug(Kroneckerfactorx);
```

$$Kroneckerfactorx \quad (8.3.85.6)$$

```
> Kroneckerfactorx(6*x^2+12*x+12);
```

$$[2, 3, x^2 + 2\,x + 2] \quad (8.3.85.7)$$

->8.3.86. Feladat.

->8.3.87. Feladat.

->8.3.88. Feladat.

->8.3.89. Feladat.

*8.3.90. Cardano-képlet.

```
> p:='p'; q:='q'; solve(x^3+p*x+q=0,x);
```

$$p := p$$

$$q := q$$

$$\frac{1}{6}\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}-\frac{2\,p}{\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}},$$
$$-\frac{1}{12}\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}+\frac{p}{\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}}+\frac{1}{2}\,\mathrm{I}\sqrt{3}\left(\frac{1}{6}\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}+\frac{2\,p}{\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}}\right),$$
$$-\frac{1}{12}\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}+\frac{p}{\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}}-\frac{1}{2}\,\mathrm{I}\sqrt{3}\left(\frac{1}{6}\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}+\frac{2\,p}{\left(-108\,q+12\sqrt{12\,p^3+81\,q^2}\right)^{1/3}}\right) \qquad (8.3.90.1)$$

```
> r:='r'; solve(x^4+p*x^2+q*x+r=0,x);
```

$$r := r$$

$$\mathrm{RootOf}\left(\_Z^4+p\,\_Z^2+q\,\_Z+r\right) \qquad (8.3.90.2)$$

▼ ***8.3.91. Feladat.***

▼ ***8.3.92. Feladat.***

▶ ***8.3.93. Testbővítések.***

▶ ***8.3.94. Tétel.***

▶ ***8.3.95. Feladat: testbővítések.***

▼ ***8.3.96. Feladat.***

▶ ***8.3.97. Feladat.***

▼ ***8.3.98. Feladat.***

▼ ***8.3.99. Feladat.***

8.3.100. Feladat.

8.3.101. Feladat.

8.3.102. Feladat.

8.3.103. Feladat.

8.3.104. Feladat: szerkeszthetőség.

8.3.105. Feladat.

8.3.106. Feladat.

8.3.107. Feladat.

8.3.108. Feladat.

8.3.109. Feladat.

8.3.110. Véges testek alaptétele.

8.3.111. Wedderburn tétele.

*8.3.112. Polinomfaktorizálás véges testek felett.

*8.3.113. Magasabb fokú kongruenciák.

*8.3.114. Feladat.

*8.3.115. Hensel-lemma.

*8.3.116. Megjegyzés.

8.3.117. Feladat.

*8.3.118. Feladat.

8.3.119. Feladat.

*8.3.120. Feladat.

8.3.121. Racionális törtfüggvények.

```
> r1:=(z^3-1)/(z^2-1); r1:=simplify(r1); r2:=simplify((z^4-1)
  /(z^3-1));
  r1*r2;
```

$$r1 := \frac{z^3 - 1}{z^2 - 1}$$

$$r1 := \frac{z^2 + z + 1}{z + 1}$$

$$r2 := \frac{z^3 + z^2 + z + 1}{z^2 + z + 1}$$

$$\frac{z^3 + z^2 + z + 1}{z + 1} \tag{8.3.121.1}$$

->8.3.122. Feladat.

8.3.123. *Parciális törtekre bontás.*

```
> f:=1/(x^4-x^2); convert(f,parfrac,x);
```

$$f := \frac{1}{x^4 - x^2}$$

$$-\frac{1}{x^2} + \frac{1}{2\,(x-1)} - \frac{1}{2\,(x+1)} \qquad (8.3.123.1)$$

8.3.124. *Következmény.*

```
> f:=36/(x^5-2*x^4-2*x^3+4*x^2+x-2); convert(f,parfrac,x);
```

$$f := \frac{36}{x^5 - 2\,x^4 - 2\,x^3 + 4\,x^2 + x - 2}$$

$$-\frac{3}{(x+1)^2} - \frac{4}{x+1} - \frac{9}{(x-1)^2} + \frac{4}{x-2} \qquad (8.3.124.1)$$

8.3.125. *Következmény.*

```
> f:=(x^5+1)/(x^4-x^2); convert(f,parfrac,x);
```

$$f := \frac{x^5 + 1}{x^4 - x^2}$$

$$x - \frac{1}{x^2} + \frac{1}{x-1} \qquad (8.3.125.1)$$

8.3.126. *Megjegyzés.*

```
> b:='b'; f:=x/(x-b)^2; convert(f,parfrac,x);
```

$$b := b$$

$$f := \frac{x}{(x-b)^2}$$

$$\frac{b}{(x-b)^2} + \frac{1}{x-b} \qquad (8.3.126.1)$$

```
> f:=(4*x^3-6*x^2-2)/(x^4-2*x^3-2*x+4); convert(f,parfrac,x);
```

$$f := \frac{4\,x^3 - 6\,x^2 - 2}{x^4 - 2\,x^3 - 2\,x + 4}$$

$$\frac{3\,x^2}{x^3 - 2} + \frac{1}{x-2} \qquad (8.3.126.2)$$

```
> convert(f,parfrac,x,2^(1/3));
```

(8.3.126.3)

$$-\frac{\left(-1+2^{2/3}\right)2^{1/3}}{\left(x-2^{1/3}\right)\left(-2+2^{1/3}\right)} - \frac{6}{(x-2)\left(4+2\,2^{1/3}+2^{2/3}\right)\left(-2+2^{1/3}\right)} + \frac{-2\,x-2^{1/3}}{-x^2-x\,2^{1/3}-2^{2/3}} \quad (8.3.126.3)$$

->8.3.127. Feladat.

8.3.128. Többhatározatlanú polinomok.

```
> p:=3*x^3*y^2+4*x^3*y+7*x^2*y^2*z+9; indets(p); coeff(p,x^3)
  ; coeff(p,y^2);
  degree(p,x); degree(p,{x,z}); degree(p);
```

$$p := 3\,x^3\,y^2 + 4\,x^3\,y + 7\,x^2\,y^2\,z + 9$$

$$\{x, z, y\}$$

$$3\,y^2 + 4\,y$$

$$3\,x^3 + 7\,x^2\,z$$

$$3$$

$$3$$

$$5 \quad (8.3.128.1)$$

```
>
>
```

*8.3.129. Multiindexek.

*8.3.130. Formális hatványsorok.

8.3.131. Tétel.

8.3.132. Megjegyzés.

*8.3.133. Szimmetrikus polinomok.

*8.3.134. Szimmetrikus polinomok alaptétele.

*8.3.135. Newton képletei.

->8.3.136. Feladat.

*8.3.137. Feladat.

*8.3.138. Feladat.

*8.3.139. Feladat.

*8.3.140. Feladat.

*8.3.141. Feladat.

8.3.142. További feladatok megoldásokkal.

8.3.143. További feladatok.

▶ 9. Kódolás

▶ 10. Algoritmusok