

Illustration der Arithmetik in den Ringen \mathbb{Z}_n und ihren Einheitengruppen \mathbb{Z}_n^*

Laden des Maple-Paketes *numtheory*, das u.a. die Funktionen *phi* (Eulersche phi-Funktion) und *order* (Ordnung eines Elementes modulo n) zur Verfügung stellt

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

```
[GIgcd, bigomega, cfrac, cfracpol, cyclotomic, divisors, factorEQ, factorset, fermat,
imagunit, index, integral_basis, invcfrac, invphi, issqrfree, jacobi, kronecker, lambda,
legendre, mcombine, mersenne, minkowski, mipolys, mlog, mobius, mroot, msqrt,
nearestp, nthconver, nthdenom, nthnumer, nthpow, order, pdexpand, phi, pi,
pprimroot, primroot, quadres, rootsunity, safeprime, sigma, sq2factor, sum2sqr, tau,
thue]
```

```
> coprime := proc(a,b)
> #
> # Test auf Teilerfremdheit
> #
> if igcd(a,b)=1 then 'true' else 'false' fi;
> end;
```

```
coprime := proc(a, b) if igcd(a, b) = 1 then 'true' else 'false' end ifend proc
```

Addition (= additive Gruppenstruktur von \mathbb{Z}_n)

```
> createadd := proc (n)
> #
> # Additionstabelle von  $\mathbb{Z}_n$ 
> #
> matrix(n,n,(x,y) -> x+y-2 mod n);
> end;
```

```
createadd := proc(n) matrix(n, n, (x, y) -> (x + y - 2) mod n) end proc
```

```
> createadd(5);
```

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

```
> createadd(6);
```

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 & 0 \\ 2 & 3 & 4 & 5 & 0 & 1 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 4 & 5 & 0 & 1 & 2 & 3 \\ 5 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

```
> createadd(12);
```

0	1	2	3	4	5	6	7	8	9	10	11
1	2	3	4	5	6	7	8	9	10	11	0
2	3	4	5	6	7	8	9	10	11	0	1
3	4	5	6	7	8	9	10	11	0	1	2
4	5	6	7	8	9	10	11	0	1	2	3
5	6	7	8	9	10	11	0	1	2	3	4
6	7	8	9	10	11	0	1	2	3	4	5
7	8	9	10	11	0	1	2	3	4	5	6
8	9	10	11	0	1	2	3	4	5	6	7
9	10	11	0	1	2	3	4	5	6	7	8
10	11	0	1	2	3	4	5	6	7	8	9
11	0	1	2	3	4	5	6	7	8	9	10

```
> createadd(17);
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	1
3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	1	2
4	5	6	7	8	9	10	11	12	13	14	15	16	0	1	2	3
5	6	7	8	9	10	11	12	13	14	15	16	0	1	2	3	4
6	7	8	9	10	11	12	13	14	15	16	0	1	2	3	4	5
7	8	9	10	11	12	13	14	15	16	0	1	2	3	4	5	6
8	9	10	11	12	13	14	15	16	0	1	2	3	4	5	6	7
9	10	11	12	13	14	15	16	0	1	2	3	4	5	6	7	8
10	11	12	13	14	15	16	0	1	2	3	4	5	6	7	8	9
11	12	13	14	15	16	0	1	2	3	4	5	6	7	8	9	10
12	13	14	15	16	0	1	2	3	4	5	6	7	8	9	10	11
13	14	15	16	0	1	2	3	4	5	6	7	8	9	10	11	12
14	15	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13
15	16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
> createadd(18);
```

$$\begin{bmatrix} 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 \\ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0 \\ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1 \\ 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2 \\ 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3 \\ 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4 \\ 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5 \\ 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6 \\ 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7 \\ 9, 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8 \\ 10, 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \\ 11, 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \\ 12, 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 \\ 13, 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \\ 14, 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 \\ 15, 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 \\ 16, 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 \\ 17, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 \end{bmatrix}$$

Multiplikation (multiplikative Halbgruppenstruktur von Z_n)

```

> createmult := proc (n)
> #
> # Multiplikationstabelle von  $Z_n$ 
> #
> matrix(n,n,(x,y) -> (x-1)*(y-1) mod n);
> end;
createmult := proc(n) matrix(n, n, (x, y) -> (x - 1) * (y - 1) mod n) end proc
> createmult(5);

```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 1 & 3 \\ 0 & 3 & 1 & 4 & 2 \\ 0 & 4 & 3 & 2 & 1 \end{bmatrix}$$

```

> createmult(6);

```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 0 & 4 & 2 & 0 & 4 & 2 \\ 0 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

```

> createmult(12);

```

```

[ 0 0 0 0 0 0 0 0 0 0 0 0 ]
[ 0 1 2 3 4 5 6 7 8 9 10 11 ]
[ 0 2 4 6 8 10 0 2 4 6 8 10 ]
[ 0 3 6 9 0 3 6 9 0 3 6 9 ]
[ 0 4 8 0 4 8 0 4 8 0 4 8 ]
[ 0 5 10 3 8 1 6 11 4 9 2 7 ]
[ 0 6 0 6 0 6 0 6 0 6 0 6 ]
[ 0 7 2 9 4 11 6 1 8 3 10 5 ]
[ 0 8 4 0 8 4 0 8 4 0 8 4 ]
[ 0 9 6 3 0 9 6 3 0 9 6 3 ]
[ 0 10 8 6 4 2 0 10 8 6 4 2 ]
[ 0 11 10 9 8 7 6 5 4 3 2 1 ]

```

```
> createmult(17);
```

```

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ]
[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ]
[ 0 2 4 6 8 10 12 14 16 1 3 5 7 9 11 13 15 ]
[ 0 3 6 9 12 15 1 4 7 10 13 16 2 5 8 11 14 ]
[ 0 4 8 12 16 3 7 11 15 2 6 10 14 1 5 9 13 ]
[ 0 5 10 15 3 8 13 1 6 11 16 4 9 14 2 7 12 ]
[ 0 6 12 1 7 13 2 8 14 3 9 15 4 10 16 5 11 ]
[ 0 7 14 4 11 1 8 15 5 12 2 9 16 6 13 3 10 ]
[ 0 8 16 7 15 6 14 5 13 4 12 3 11 2 10 1 9 ]
[ 0 9 1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 ]
[ 0 10 3 13 6 16 9 2 12 5 15 8 1 11 4 14 7 ]
[ 0 11 5 16 10 4 15 9 3 14 8 2 13 7 1 12 6 ]
[ 0 12 7 2 14 9 4 16 11 6 1 13 8 3 15 10 5 ]
[ 0 13 9 5 1 14 10 6 2 15 11 7 3 16 12 8 4 ]
[ 0 14 11 8 5 2 16 13 10 7 4 1 15 12 9 6 3 ]
[ 0 15 13 11 9 7 5 3 1 16 14 12 10 8 6 4 2 ]
[ 0 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 ]

```

```
> createmult(18);
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	2	4	6	8	10	12	14	16	0	2	4	6	8	10	12	14	16
0	3	6	9	12	15	0	3	6	9	12	15	0	3	6	9	12	15
0	4	8	12	16	2	6	10	14	0	4	8	12	16	2	6	10	14
0	5	10	15	2	7	12	17	4	9	14	1	6	11	16	3	8	13
0	6	12	0	6	12	0	6	12	0	6	12	0	6	12	0	6	12
0	7	14	3	10	17	6	13	2	9	16	5	12	1	8	15	4	11
0	8	16	6	14	4	12	2	10	0	8	16	6	14	4	12	2	10
0	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9	0	9
0	10	2	12	4	14	6	16	8	0	10	2	12	4	14	6	16	8
0	11	4	15	8	1	12	5	16	9	2	13	6	17	10	3	14	7
0	12	6	0	12	6	0	12	6	0	12	6	0	12	6	0	12	6
0	13	8	3	16	11	6	1	14	9	4	17	12	7	2	15	10	5
0	14	10	6	2	16	12	8	4	0	14	10	6	2	16	12	8	4
0	15	12	9	6	3	0	15	12	9	6	3	0	15	12	9	6	3
0	16	14	12	10	8	6	4	2	0	16	14	12	10	8	6	4	2
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Inverse in Z_n

Test auf Invertierbarkeit in Z_n : *igcd* ist Maple-Funktion zur Berechnung des ggT fuer ganze Zahlen "*" bedeutet: Inverses existiert nicht in Z_n

```

> checkinverse := proc (n)
> #
> # Feststellen der invertierbaren Elemente in  $Z_n$ 
> #
> local inv,z;
> inv := matrix(2,n);
> for z from 1 to n do
> inv[1,z] := z-1;
> if igcd(z-1,n)=1 then inv[2,z] := (z-1)^(-1) mod n else inv[2,z]
:=
> '*' fi;
> od;
> op(inv);
> end;

```

```

checkinverse := proc(n)
local inv, z;
  inv := matrix(2, n);
  for z to n do
    inv1,z := z - 1;
    if gcd(z - 1, n) = 1 then inv2,z := 1/(z - 1) mod n else inv2,z := '*' end if
  end do;
  op(inv)
end proc
> checkinverse(5);
      [ 0  1  2  3  4 ]
      [  1  3  2  4 ]
> checkinverse(6);
      [ 0  1  2  3  4  5 ]
      [  1  *  *  *  5 ]
> checkinverse(12);
      [ 0  1  2  3  4  5  6  7  8  9  10  11 ]
      [  1  *  *  *  5  *  7  *  *  *  11 ]
> checkinverse(17);
      [ 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 ]
      [  1  9  6  13  7  3  5  15  2  12  14  10  4  11  8  16 ]
> checkinverse(18);
      [ 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 ]
      [  1  *  *  *  11  *  13  *  *  *  5  *  7  *  *  *  17 ]

```

Die Einheitengruppe der invertierbaren Elemente

Gruppenstruktur bezüglich der Multiplikation auf den invertierbaren Elementen Z_n^* (Anzahl = $\phi(n)$)

```

> createunits := proc (n)
> #
> # Elemente der Einheitengruppe  $Z_n^*$  von  $Z_n$ 
> #
> select(x -> coprime(x,n), [seq(k,k=1..n-1)]);
> end;
createunits := proc(n) select(x -> coprime(x, n), [seq(k, k = 1..n - 1)]) end proc
> createunits(5);
      [1, 2, 3, 4]
> createunits(6);
      [1, 5]
> createunits(12);

```

```

                                [1, 5, 7, 11]
> createunits(17);
                                [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
> createunits(18);
                                [1, 5, 7, 11, 13, 17]
> createunitgroup := proc (n)
> #
> # Multiplikationstabelle der Einheitengruppe modulo n
> #
> local u;
> u := createunits(n);
> matrix(phi(n),phi(n), (x,y) -> (op(x,u)*op(y,u) mod n));
> end;

createunitgroup := proc(n)
local u;
    u := createunits(n); matrix(phi(n), phi(n), (x, y) -> op(x, u) * op(y, u) mod n)
end proc
> createunitgroup(5);
                                
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

> createunitgroup(6);
                                
$$\begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix}$$

> createunitgroup(12);
                                
$$\begin{bmatrix} 1 & 5 & 7 & 11 \\ 5 & 1 & 11 & 7 \\ 7 & 11 & 1 & 5 \\ 11 & 7 & 5 & 1 \end{bmatrix}$$

> createunitgroup(17);

```

```

1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
2  4  6  8 10 12 14 16  1  3  5  7  9 11 13 15
3  6  9 12 15  1  4  7 10 13 16  2  5  8 11 14
4  8 12 16  3  7 11 15  2  6 10 14  1  5  9 13
5 10 15  3  8 13  1  6 11 16  4  9 14  2  7 12
6 12  1  7 13  2  8 14  3  9 15  4 10 16  5 11
7 14  4 11  1  8 15  5 12  2  9 16  6 13  3 10
8 16  7 15  6 14  5 13  4 12  3 11  2 10  1  9
9  1 10  2 11  3 12  4 13  5 14  6 15  7 16  8
10 3 13  6 16  9  2 12  5 15  8  1 11  4 14  7
11 5 16 10  4 15  9  3 14  8  2 13  7  1 12  6
12 7  2 14  9  4 16 11  6  1 13  8  3 15 10  5
13 9  5  1 14 10  6  2 15 11  7  3 16 12  8  4
14 11  8  5  2 16 13 10  7  4  1 15 12  9  6  3
15 13 11  9  7  5  3  1 16 14 12 10  8  6  4  2
16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1

```

```
> createunitgroup(18);
```

```

[ 1  5  7 11 13 17 ]
[ 5  7 17  1 11 13 ]
[ 7 17 13  5  1 11 ]
[11  1  5 13 17  7 ]
[13 11  1 17  7  5 ]
[17 13 11  7  5  1 ]

```

Ordnung von Elementen in Z_n^* und Nachpruefen der Aussage des Satzes von Euler

```

> eulercheck := proc (n)
> #
> # Tabelle der Potenzen z^i mod n fuer 1 <= i <= phi(n)
> #
> local u;
> u := createunits(n);
> matrix(phi(n),phi(n),(x,y) -> (op(x,u)^y mod n));
> end;

eulercheck := proc(n)
local u;
  u := createunits(n); matrix(phi(n), phi(n), (x, y) -> op(x, u)^y mod n)
end proc
> eulercheck(5);

```

```

[ 1  1  1  1 ]
[ 2  4  3  1 ]
[ 3  4  2  1 ]
[ 4  1  4  1 ]

```

```
> eulercheck(6);
```

```

                                
$$\begin{bmatrix} 1 & 1 \\ 5 & 1 \end{bmatrix}$$

> eulercheck(12);
                                
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 5 & 1 & 5 & 1 \\ 7 & 1 & 7 & 1 \\ 11 & 1 & 11 & 1 \end{bmatrix}$$

> eulercheck(17);

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 15 & 13 & 9 & 1 & 2 & 4 & 8 & 16 & 15 & 13 & 9 & 1 \\ 3 & 9 & 10 & 13 & 5 & 15 & 11 & 16 & 14 & 8 & 7 & 4 & 12 & 2 & 6 & 1 \\ 4 & 16 & 13 & 1 & 4 & 16 & 13 & 1 & 4 & 16 & 13 & 1 & 4 & 16 & 13 & 1 \\ 5 & 8 & 6 & 13 & 14 & 2 & 10 & 16 & 12 & 9 & 11 & 4 & 3 & 15 & 7 & 1 \\ 6 & 2 & 12 & 4 & 7 & 8 & 14 & 16 & 11 & 15 & 5 & 13 & 10 & 9 & 3 & 1 \\ 7 & 15 & 3 & 4 & 11 & 9 & 12 & 16 & 10 & 2 & 14 & 13 & 6 & 8 & 5 & 1 \\ 8 & 13 & 2 & 16 & 9 & 4 & 15 & 1 & 8 & 13 & 2 & 16 & 9 & 4 & 15 & 1 \\ 9 & 13 & 15 & 16 & 8 & 4 & 2 & 1 & 9 & 13 & 15 & 16 & 8 & 4 & 2 & 1 \\ 10 & 15 & 14 & 4 & 6 & 9 & 5 & 16 & 7 & 2 & 3 & 13 & 11 & 8 & 12 & 1 \\ 11 & 2 & 5 & 4 & 10 & 8 & 3 & 16 & 6 & 15 & 12 & 13 & 7 & 9 & 14 & 1 \\ 12 & 8 & 11 & 13 & 3 & 2 & 7 & 16 & 5 & 9 & 6 & 4 & 14 & 15 & 10 & 1 \\ 13 & 16 & 4 & 1 & 13 & 16 & 4 & 1 & 13 & 16 & 4 & 1 & 13 & 16 & 4 & 1 \\ 14 & 9 & 7 & 13 & 12 & 15 & 6 & 16 & 3 & 8 & 10 & 4 & 5 & 2 & 11 & 1 \\ 15 & 4 & 9 & 16 & 2 & 13 & 8 & 1 & 15 & 4 & 9 & 16 & 2 & 13 & 8 & 1 \\ 16 & 1 & 16 & 1 & 16 & 1 & 16 & 1 & 16 & 1 & 16 & 1 & 16 & 1 & 16 & 1 \end{bmatrix}$$

> eulercheck(18);
                                
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 7 & 17 & 13 & 11 & 1 \\ 7 & 13 & 1 & 7 & 13 & 1 \\ 11 & 13 & 17 & 7 & 5 & 1 \\ 13 & 7 & 1 & 13 & 7 & 1 \\ 17 & 1 & 17 & 1 & 17 & 1 \end{bmatrix}$$

> orderlist := proc (n)
> #
> # Tabelle der Ordnungen von z modulo n fuer z aus Z_n^*
> #
> local u,ord,i;
> u := createunits(n);
> ord := matrix(2,phi(n));
> for i from 1 to phi(n) do
> ord[1,i] := op(i,u);
> ord[2,i] := order(op(i,u),n)
> od;
> op(ord);
> end;

```

```

orderlist := proc(n)
local u, ord, i;
  u := createunits(n);
  ord := matrix(2, phi(n));
  for i to phi(n) do ord1,i := op(i, u); ord2,i := order(op(i, u), n) end do;
  op(ord)
end proc
> orderlist(5);
      [ 1 2 3 4 ]
      [ 1 4 4 2 ]
> orderlist(6);
      [ 1 5 ]
      [ 1 2 ]
> orderlist(12);
      [ 1 5 7 11 ]
      [ 1 2 2 2 ]
> orderlist(17);
      [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ]
      [ 1 8 16 4 16 16 16 8 8 16 16 16 4 16 8 2 ]
> orderlist(18);
      [ 1 5 7 11 13 17 ]
      [ 1 6 3 6 3 2 ]

```

Werte der Eulerschen phi-Funktion

```

> philist := proc (n)
> #
> # Tabelle der Werte der Eulerschen phi-Funktion fuer 1 <= i <=
> n
> #
> local phimat,i;
> phimat := matrix(2,n);
> for i from 1 to n do
> phimat[1,i] := i;
> phimat[2,i] := phi(i)
> end;
> op(phimat);
> end;

```

```

    philist := proc(n)
    local phimat, i;
        phimat := matrix(2, n);
        for i to n do phimat1, i := i; phimat2, i :=  $\phi(i)$  end do;
        op(phimat)
    end proc
> philist(20);

```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 1 & 1 & 2 & 2 & 4 & 2 & 6 & 4 & 6 & 4 & 10 & 4 & 12 & 6 & 8 & 8 & 16 & 6 & 18 & 8 \end{bmatrix}$$