Modulo-Rechnen, Zahlentheorie

Mathematik in wxMaxima www.mathematik-verstehen.de Haftendorn Okt 2010

Hilfen zum Handling
Achtung: Durch Anklicken der linken Zellmarkierung kann man die
Abschnitte und auch einzelne Zellen aufklappen und auch wieder zuklappen.
Dazu Shift halten, dann werden auch alle Unterebenen aufgeklappt.
Endung *.wxmx ist komfortabel. Ist die Endung *.wxm muss man
erst noch alle Ausgaben neu erzeugen. Mit Strg r werden alle aufgeklappten
Zum Lernen ist es besser die Zellen einzeln (mit Shift+Enter) auszuwerten.
Werte einzelne Zellen aus mit Shift-Enter.
Auswertung in einem Rutsch: Falte alle Abschnitte auf,
werte alle Zellen mit Strg r aus (auch Menu Cell Alle Zellen auswerten).

Figure 1: Inhaltsverzeichnis

1 Zahlen, Teilbakkeit, Vielfache

1.1 Ganze Zahlen, Ganzzahlige Division

In der Zahlentheorie kommen nur die Ganzen Zahlen vor.
Dezimalzahlen spielen keine Rolle.
Bei Divisionen interessiert man sich für den ganzzahlichen Anteil und den Re

```
(%i1) 34/7;
      divide(34,7);
(%o1) 34/7
(%o2) [4,6]
```

also 34 geteilt durch 7 ist 4 Ganze Rest 7 (wie in der Grundschule)

```
(%i3) num(34/7);
      denom(34/7);
(%o3) 34
(%o4) 7
```

Zähler (numerator) und Nenner (denominator) kann man aus
Brüchen herausgreifen.

1.2 Primfaktoren und Teiler

```
(%i5) factor(731);
      divisors(731);
(%o5) 17 43
(%o6) {1,17,43,731}
```

```
(%i7) factor(72);
      ifactors(72);
      divisors(72);
(%o7) 2^3 3^2
(%o8) [[2,3],[3,2]]
(%o9) {1,2,3,4,6,8,9,12,18,24,36,72}
```

```
(%i10) factor(123456789);
(%o10) 3^2 3607 3803
```

```
Die Zerlegung in Faktoren zeigt die Primfaktoren und ihre Potenzen.
Bei ifactors werden diese als Liste von Primahlen mit ihren Potenzen
ausgegeben.  (integer =ganzzahlig)
Die Teiler einer Zahl sind alle Produkte, sich mit diesen Bausteinen
bilden lassen. Z.B. ist 3*3607 Teiler, aber auch 3^2*3803
```

```
(%i11) 3*3607; 3^2*3803;
(%o11) 10821
(%o12) 34227
```

```
(%i13) divide(123456789,10821);  divide(123456789,34227);
(%o13) [11409,0]
(%o14) [3607,0]
```

```
Der Rest 0 war also vorherzusehen.
```

```
der Befehl divisors(n) zeigt die Teilemenge von n.
```

```
(%i15) divisors(3607);
(%o15) {1,3607}
```

```
Die Zahlen, die nur sich und die 1 als Teiler haben, heißen Primzahlen.
3607 ist also eine Primzahl.
Achtung: 1 ist keine Primzahl, das ist so definiert.
```

1.3 Gemeinsame Teiler, ggT

```
(%i16) divisors(72); divisors(48);
(%o16) {1,2,3,4,6,8,9,12,18,24,36,72}
(%o17) {1,2,3,4,6,8,12,16,24,48}
```

```
Der größte gemeinsame Teiler ist ersichtlich 24.
Dafür gibt es den Begriff ggT(72,48) in Deutsch und in Eglisch
gcd(72,48)  greatest common divisor
```

```
(%i18) gcd(72,48);
(%o18) 24

(%i19) divisors(24);
(%o19) {1,2,3,4,6,8,12,24}
```

In dieser Teilermenge sind tatsächlich alle gemeinsamen Elemente von den obigen beiden Teilermengen und 24 ist die größte unter ihnen.

Es gibt den Euklidischen Algorithmus auch in seiner erweiterten Form

```
(%i20) gcdex(72,48);
(%o20) [1,-1,24]
```

gcdex(a,b) ist so zu lesen: [r,s,gcd] mit r*a+s*b=gcd(a,b)
Also hier

```
(%i21) 1*72+(-1)*48=24;
(%o21) 24 =24
```

Diese "Vielfachsummendarstellung" wird in der Kryptografie sehr wichtig.
An passender Stelle wird dies aufgegriffen.
Achtung andere Software schreibt [gcd,r,s].

1.4 Vielfache und Gemeinsame Vielfache

Die Vielfachenmengen sind naturgemäß unendlich groß.
Hier nehmen wir 10 Elemente. Wir bleiben im Positiven, damit es übersichtlich

```
(%i22) Z:makelist(i,i,0,10);
(%o22) [0,1,2,3,4,5,6,7,8,9,10]
```

Erstmal erzeugen wir und die Grundmenge Z (eigentlich noch mit Negativen)

```
(%i23) 6*Z;
(%o23) [0,6,12,18,24,30,36,42,48,54,60]
```

Das ist die abgekürzte Vielfachenmenge 6 Z als Liste.

```
(%i24) 8*Z;
(%o24) [0,8,16,24,32,40,48,56,64,72,80]
```

Unter den gemeinsamen Elementen von 6Z und 8Z ist 24 das kleinste
kgV(6,8)=24 in Deutsch, in Englisch lcm(6,8) least common multiple

```
(%i25) lcm(6,8);
Warning - you are redefining the Maxima function lcm
(%o25) 24
```

2 Die Gruppe Z(m) und das Modulo-Rechnen

2.1 modulo - Begriff, Z(m)

Es kommt hier nur auf die Reste an, die beim ganzzahligen Teilen bleiben.
In mathematischer Schreibweise 34 mod 7 = 6 (lies 34 modulo 7 ist 6)

```
(%i26) mod(34,7);
(%o26) 6
```

Nehmen wir von allen ganzen Zahlen immer nur den Rest modulo 7
so erhalten wir offfenbar eine sehr kleine Menge, sie heißt Z(7)

```
(%i27) Z;
(%o27) [0,1,2,3,4,5,6,7,8,9,10]
```

```
(%i28) mod(Z,7);
       unique(mod(Z,7));
(%o28) [0,1,2,3,4,5,6,0,1,2,3]
(%o29) [0,1,2,3,4,5,6]
```

Jetzt nehmen wir negative Zahlen hinzu

```
(%i30) ZZ:  makelist(i,i,-10,20);
(%o30) [-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
17,18,19,20]
```

```
(%i31) mod(ZZ,7);
       unique(mod(ZZ,7));
(%o31) [4,5,6,0,1,2,3,4,5,6,0,1,2,3,4,5,6,0,1,2,3,4,5,6,0,1,2,3,4,5,6]
(%o32) [0,1,2,3,4,5,6]
```

unique wirft die Doppelungen weg. Wir definieren also:

```
(%i33) Z(m__):=unique(mod(ZZ,m__))$
```

```
(%i34) Z(3); Z(7); Z(16);
(%o34) [0,1,2]
(%o35) [0,1,2,3,4,5,6]
(%o36) [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
```

2.2 Rechnen in Z(m)

Mit den Zahlen dieser Menge kann man nun die Grundrechenarten
+,-, *, hoch unbeschränkt ausführen, "geteilt" gehört nicht dazu.
Grundprinzip ist, dass man wie in ganzen Zahlen rechnet und alles modulo m
interpretiert.

```
(%i37) 11+13;
       mod(11+13,16);
(%o37) 24
(%o38) 8
```

```
(%i39) 11-13;
       mod(11-13,16);
(%o39) -2
(%o40) 14
```

```
Wenn man das selbst rechnen will, zählt man zu der -2 eine 16 hinzu, ist 14.
Oder man denkt 11=27 mod 16  und  27-13 =14
```

```
(%i41) mod(11,16);  mod(27,16);
(%o41) 11
(%o42) 11
```

```
Man kann an jeder Stelle des Rechenvorgangs modulo m "herunternbrechen.
Das ist besonders interessant beim Potenzieren
```

```
(%i43) mod(3^5,10);
(%o43) 3
```

```
selber: 3^2 =-1 mod 10 , 3^4=(3^2)^2=(-1)^2 mod 10 =1 mod 10;
3^5=1*3 mod 10=3
```

```
(%i44) 3^5;
(%o44) 243
```

```
Da sieht man die 3 auch gleich als Rest beim Teilen durch 10.
```

2.3 Multplikationstafeln von Z(m)

```
(%i45) /*Definitionen, nach Auwertung wieder zuklappen */
       plus(i,j,m):=mod(Z(m)[i]+Z(m)[j],m)$
       maal(i,j,m):=mod(Z(m)[i]*Z(m)[j],m)$
       plustafel(m):=block(
          array(ma,fixnum,m,m),
          for i:1 thru m do for j:1 thru m do ma[i,j]:plus(i,j,m),
          print("Plus-Tafel  modulo ",m),
          genmatrix(ma,m,m)
       )$
       maltafel(m):=block(
          array(ma,fixnum,m-1,m-1),
          for i:1 thru m-1 do for j:1 thru m-1 do ma[i,j]:maal(i+1,j+1,m),
          print("Mal-Tafel  modulo ",m),
          genmatrix(ma,m-1,m-1)
       )$
```

```
(%i49) m:6;Z(m);Z(m)[3];
(%o49) 6
(%o50) [0,1,2,3,4,5]
(%o51) 2
```

```
(%i52) plustafel(6);
```

Plus-Tafel modulo 6

$$(\%o52)\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 & 0 \\ 2 & 3 & 4 & 5 & 0 & 1 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 4 & 5 & 0 & 1 & 2 & 3 \\ 5 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

```
(%i53) maltafel(6);
```

Mal-Tafel modulo 6

$$(\%o53)\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 0 & 2 & 4 \\ 3 & 0 & 3 & 0 & 3 \\ 4 & 2 & 0 & 4 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

```
(%i54) for i:4 thru 11 do print(maltafel(i))$
```

Mal-Tafel modulo 4

1	2	3
2	0	2
3	2	1

Mal-Tafel modulo 5

1	2	3	4
2	4	1	3
3	1	4	2
4	3	2	1

Mal-Tafel modulo 6

1	2	3	4	5
2	4	0	2	4
3	0	3	0	3
4	2	0	4	2
5	4	3	2	1

Mal-Tafel modulo 7

1	2	3	4	5	6
2	4	6	1	3	5
3	6	2	5	1	4
4	1	5	2	6	3
5	3	1	6	4	2
6	5	4	3	2	1

Mal-Tafel modulo 8

1	2	3	4	5	6	7
2	4	6	0	2	4	6
3	6	1	4	7	2	5
4	0	4	0	4	0	4
5	2	7	4	1	6	3
6	4	2	0	6	4	2
7	6	5	4	3	2	1

Mal-Tafel modulo 9

1	2	3	4	5	6	7	8
2	4	6	8	1	3	5	7
3	6	0	3	6	0	3	6
4	8	3	7	2	6	1	5
5	1	6	2	7	3	8	4
6	3	0	6	3	0	6	3
7	5	3	1	8	6	4	2
8	7	6	5	4	3	2	1

Mal-Tafel modulo 10

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

2.4 Zstern(m) die Gruppe der Teilerfremden

Teilerfremd zu m sind alle Zahlen i mit ggT(m,i)=1

```
(%i55) /*Definitionen Zstern(m) und malsterntafel(m)*/
       Zstern(m):=block([li],
          li:[],
          for i:1 thru m-1 do
             if gcd(m,i)=1 then li:endcons(i,li),
          return(li)
       )$
       malsterntafel(m):=block([le,Zs,malstern],
          Zs:Zstern(m), le:length(Zs),
          malstern(i,j,m):=(mod(Zs[i]*Zs[j],m)),
          array(ma,fixnum,le,le),
          for i:1 thru le do for j:1 thru le do ma[i,j]:malstern(i,j,m),
          print("Malstern-Tafel  modulo ",m),
          genmatrix(ma,le,le)
       )$
```

```
(%i57) Zstern(20);
(%o57) [1,3,7,9,11,13,17,19]
```

Für die Anzahl dieser Teilerfremden gibt es die Eulersche-Phi-Funktion

```
(%i58) eulerphi(m):=length(Zstern(m))$
       eulerphi(20);
(%o59) 8
```

```
(%i60) malsterntafel(20);
```

Malstern-Tafel modulo 20

(%o60)

1	3	7	9	11	13	17	19
3	9	1	7	13	19	11	17
7	1	9	3	17	11	19	13
9	7	3	1	19	17	13	11
11	13	17	19	1	3	7	9
13	19	11	17	3	9	1	7
17	11	19	13	7	1	9	3
19	17	13	11	9	7	3	1

```
(%i61) for i:6 thru 21 do print(malsterntafel(i))$
```

Malstern-Tafel modulo 6

1	5
5	1

Malstern-Tafel modulo 7

1	2	3	4	5	6
2	4	6	1	3	5
3	6	2	5	1	4
4	1	5	2	6	3
5	3	1	6	4	2
6	5	4	3	2	1

Malstern-Tafel modulo 8

1	3	5	7
3	1	7	5
5	7	1	3
7	5	3	1

Malstern-Tafel modulo 9

1	2	4	5	7	8
2	4	8	1	5	7
4	8	7	2	1	5
5	1	2	7	8	4
7	5	1	8	4	2
8	7	5	4	2	1

Malstern-Tafel modulo 10

1	3	7	9
3	9	1	7
7	1	9	3
9	7	3	1

Malstern-Tafel modulo 11

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

Malstern-Tafel modulo 12

1	5	7	11
5	1	11	7
7	11	1	5

2.5 Potenztafeln

```
(%i62) /* Definition potenztafel(m) */
       potenztafel(m):=block([le,Zs,hochstern],
          Zs:Zstern(m), le:length(Zs),
          hochstern(i,j,m):=(mod(Zs[j]^i,m)),
          array(ma,fixnum,le,le),
          for i:1 thru le do for j:1 thru le do ma[i,j]:hochstern(i,j,m),
          print("Potenz-Tafel von Zstern  modulo ",m),
          print("Zstern(",m,") hat ",le," Elemente"),
          genmatrix(ma,le,le)
       )$
```

```
(%i63) potenztafel(7);
```

Potenz-Tafel von Zstern modulo 7

Zstern(7 *) hat* 6 *Elemente*

(%o63)

1	2	3	4	5	6
1	4	2	2	4	1
1	1	6	1	6	6
1	2	4	4	2	1
1	4	5	2	3	6
1	1	1	1	1	1

```
(%i64) for i:6 thru 21 do print(potenztafel(i))$
```

Potenz-Tafel von Zstern modulo 6
Zstern(6 *) hat* 2 *Elemente*

1	5
1	1

Potenz-Tafel von Zstern modulo 7
Zstern(7 *) hat* 6 *Elemente*

1	2	3	4	5	6
1	4	2	2	4	1
1	1	6	1	6	6
1	2	4	4	2	1
1	4	5	2	3	6
1	1	1	1	1	1

Potenz-Tafel von Zstern modulo 8
Zstern(8 *) hat* 4 *Elemente*

1	3	5	7
1	1	1	1
1	3	5	7
1	1	1	1

Potenz-Tafel von Zstern modulo 9
Zstern(9 *) hat* 6 *Elemente*

1	2	4	5	7	8
1	4	7	7	4	1
1	8	1	8	1	8
1	7	4	4	7	1
1	5	7	2	4	8
1	1	1	1	1	1

Potenz-Tafel von Zstern modulo 10
Zstern(10 *) hat* 4 *Elemente*

1	3	7	9
1	9	9	1
1	7	3	9
1	1	1	1

Potenz-Tafel von Zstern modulo 11
Zstern(11 *) hat* 10 *Elemente*

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

(%i65) potenztafel(30);potenztafel(31);

Potenz-Tafel von Zstern modulo 30

Zstern(30 *) hat* 8 *Elemente*

(%o65)

1	7	11	13	17	19	23	29
1	19	1	19	19	1	19	1
1	13	11	7	23	19	17	29
1	1	1	1	1	1	1	1
1	7	11	13	17	19	23	29
1	19	1	19	19	1	19	1
1	13	11	7	23	19	17	29
1	1	1	1	1	1	1	1

Potenz-Tafel von Zstern modulo 31

Zstern(31 *) hat* 30 *Elemente*

(%o66)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	9	16	25	5	18	2	19	7	28	20	14	10	8	8	10	14	20	28	7	19	2	18	5	25	16	9	4	1
1	8	27	2	1	30	2	16	16	8	29	23	27	16	27	4	15	4	8	2	23	15	15	29	1	30	29	4	23	30
1	16	19	8	5	25	14	4	20	18	9	28	10	7	2	2	7	10	28	9	18	20	4	14	25	5	8	19	16	1
1	1	26	1	25	26	5	1	25	25	6	26	6	5	30	1	26	25	5	25	6	6	30	26	5	6	30	5	30	30
1	2	16	4	1	1	4	8	8	2	4	2	16	8	16	16	8	16	2	4	2	8	8	4	1	1	4	16	2	1
1	4	17	16	5	6	28	2	10	20	13	24	22	19	23	8	12	9	7	18	11	21	29	3	25	26	15	14	27	30
1	8	20	2	25	5	10	16	28	14	19	9	7	18	4	4	18	7	9	19	14	28	16	10	5	25	2	20	8	1
1	16	29	8	1	30	8	4	4	16	23	15	29	4	29	2	27	2	16	8	15	27	27	23	1	30	23	2	15	30
1	1	25	1	5	25	25	1	5	5	5	25	5	25	1	1	25	5	25	5	5	5	1	25	25	5	1	25	1	1
1	2	13	4	25	26	20	8	14	19	24	21	3	9	15	16	22	28	10	7	12	17	23	11	5	6	27	18	29	30
1	4	8	16	1	1	16	2	2	4	16	4	8	2	8	8	2	8	4	16	4	2	2	16	1	1	16	8	4	1
1	8	24	2	5	6	19	16	18	9	21	17	11	28	27	4	3	20	14	10	22	13	15	12	25	26	29	7	23	30
1	16	10	8	25	5	9	4	7	28	14	18	19	20	2	2	20	19	18	14	28	7	4	9	5	25	8	10	16	1
1	1	30	1	1	30	1	1	1	1	30	30	30	1	30	1	30	1	1	1	30	30	30	30	1	30	30	1	30	30
1	2	28	4	5	25	7	8	9	10	20	19	18	14	16	16	14	18	19	20	10	9	8	7	25	5	4	28	2	1
1	4	22	16	25	26	18	2	19	7	3	11	17	10	23	8	21	14	20	28	24	12	29	13	5	6	15	9	27	30
1	8	4	2	1	1	2	16	16	8	2	8	4	16	4	4	16	4	8	2	8	16	16	2	1	1	2	4	8	1
1	16	12	8	5	6	14	4	20	18	22	3	21	7	29	2	24	10	28	9	13	11	27	17	25	26	23	19	15	30
1	1	5	1	25	5	5	1	25	25	25	5	25	5	1	1	5	25	5	25	25	25	1	5	5	25	1	5	1	1
1	2	15	4	1	30	4	8	8	2	27	29	15	8	15	16	23	16	2	4	29	23	23	27	1	30	27	16	29	30
1	4	14	16	5	25	28	2	10	20	18	7	9	19	8	8	19	9	7	18	20	10	2	28	25	5	16	14	4	1
1	8	11	2	25	26	10	16	28	14	12	22	24	18	27	4	13	7	9	19	17	3	15	21	5	6	29	20	23	30
1	16	2	8	1	1	8	4	4	16	8	16	2	4	2	2	4	2	16	8	16	4	4	8	1	1	8	2	16	1
1	1	6	1	5	6	25	1	5	5	26	6	26	25	30	1	6	5	25	5	26	26	30	6	25	26	30	25	30	30
1	2	18	4	25	5	20	8	14	19	7	10	28	9	16	16	9	28	10	7	19	14	8	20	5	25	4	18	2	1
1	4	23	16	1	30	16	2	2	4	15	27	23	2	23	8	29	8	4	16	27	29	29	15	1	30	15	8	27	30
1	8	7	2	5	25	19	16	18	9	10	14	20	28	4	4	28	20	14	10	9	18	16	19	25	5	2	7	8	1

3 Ordnung eines Elementes, Powermod

3.1 Ordnung

Wenn man sich die Potenztafeln ansieht, fällt auf, dass alle
in der letzten Zeile ausschließlich 1 zeigen.
Das passt zu dem Satz der Gruppentheorie:
Element hoch Gruppenordnung =1
Die Gruppenordnung ist die Zahl der Elemente einer Gruppe.
Die Gruppen Zstern(m) haben eulerphi(m) Elemente

```
(%i67) Zstern(10); eulerphi(10);
(%o67) [1,3,7,9]
(%o68) 4
```

```
(%i69) Zstern(16); eulerphi(16);
(%o69) [1,3,5,7,9,11,13,15]
(%o70) 8
```

Man bildet den Begriff Ordnung eines Elementes:
ord(a)=k genau wenn k die kleinste Zahl mit a^k mod m =1 ist.
In den Potenztafeln ist k die Nummer der Zeile, in der zum ersten Mal
eine Eins in der k-Spalte auftaucht.

```
(%i71) makelist(mod(7^i,16),i,1,8);
(%o71) [7,1,7,1,7,1,7,1]
```

```
(%i72) makelist(mod(5^i,16),i,1,8);
(%o72) [5,9,13,1,5,9,13,1]
```

7 hat also in Zstern(16) die Ordnung 2, 5 hat die Ordnung 4

```
(%i73) ordo(a,m):=block([p,z],
                         p:a,z:1, while p>1 do (p:mod(a*p,m), z:z+1),
                         return(z)
                        );
(%o73) ordo(a,m):=
block([p,z],p:a,z:1,while p>1 do (p:mod(a p,m),z:z+1),return(z))
```

```
(%i74) ordo(13,16);
(%o74) 4
```

```
(%i75) makelist(mod(5^(4*n),16),n,1,10);
(%o75) [1,1,1,1,1,1,1,1,1,1]
```

Man überlegt leicht, dass 5 hoch eine beliebiges 4-Vielfache
modulo 16 gleich 1 ist. Darum kann man 5^2010 mod 16 im Kopf ausrechnen.
5^2010=5^(2008+2) mod 16 = 5^2 mod 16=25 mod 16=9

```
(%i76) mod(5^2010,16);
(%o76) 9
```

Einweiter Satz der Gruppentheorie ist, dass dieElementordnung die Gruppenordnung teilen muss. (Begründung bei "Nebenklassen". Also kommen nur die Teiler von eulerphi als Elementordnungen infrage.

(%i77) potenztafel(18);

Potenz-Tafel von Zstern modulo 18

Zstern(18 *) hat* 6 *Elemente*

(%o77)
$$\begin{bmatrix} 1 & 5 & 7 & 11 & 13 & 17 \\ 1 & 7 & 13 & 13 & 7 & 1 \\ 1 & 17 & 1 & 17 & 1 & 17 \\ 1 & 13 & 7 & 7 & 13 & 1 \\ 1 & 11 & 13 & 5 & 7 & 17 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5 und 11 haben Ordnung 6, 7 und 13 haben Ordnung 3, 17 hat Ordnung2.

3.2 Powermod

Für die Kryptografie ist es wichtig, dass hohe Potenzen riesiger Zahlen berechnet werden können.

```
(%i78) /*Definition von pmod(a.k.m) powermod*/
       pmod(a,k,m):=block([x,i,pot],
          i:k, x:1,pot:a,
          marke,
          if mod(i,2)=1 then (x:mod(x*pot,m),
             if i=1 then return(x),
             i:i-1
          ),
          i:i/2,
          pot:mod(pot*pot,m),
          go(marke)
       )$
```

```
(%i79) pmod(5,3,16);mod(5^3,16);
(%o79) 13
(%o80) 13
```

```
(%i81) pmod(12345,34567,16);mod(12345^34567,16);
(%o81) 9
(%o82) 9
```

Die letzte Rechnung könnte man mit einem gewöhnlichen Taschenrechner nicht ausführen. In der Kryptografie haben die Zahlen aber etwa 200 Stellen und nicht 5 wie oben. Dann geht der rechte Befehl auch nicht meh

```
(%i83) primlistebis(n):=([i,li],li:[],for i from 1 thru n do if primep(i) then li:append(li,[i]),li);
(%o83) primlistebis(n):=
([i,li],li:[],for i thru n do if primep(i) then li:append(li,[i]) ,li)
```

```
(%i84) lll:primlistebis(500);
(%o84) [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97,
101,103,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,
199,211,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,
317,331,337,347,349,353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,
443,449,457,461,463,467,479,487,491,499]

(%i85) primep(250348280129);
(%o85) true

(%i86) makelist(pmod(99,p-1,p),p, primlistebis(500));
(%o86) [1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]

(%i87) mod(99^4998,4999);
(%o87) 1
```

3.3 Nebenklassen

```
In einer Gruppe Zstern(m) kann es Elemente geben, deren Potenzen
schon die ganze Gruppe erzeugen, nämlich die Elemente mit maximaler Ordnung.

(%i88) Zstern(18);
(%o88) [1,5,7,11,13,17]

(%i89) makelist(mod(5^i,18),i,1,6);
(%o89) [5,7,17,13,11,1]

(%i90) pl[13]:makelist(mod(13^i,18),i,1,3);
(%o90) [13,7,1]

Die Elemente, die nicht in der Potenzenliste vorkommen, bilden
mit pl[13] echte Nebenklassen  g*pl[a]

(%i91) mod(5*pl[13],18);mod(11*pl[13],18); mod(17*pl[13],18);
(%o91) [11,17,5]
(%o92) [17,5,11]
(%o93) [5,11,17]

davon stimmen einige, hier sogar alle, überein.
Das Element der Ordnung 3 hat hier nur 2 Nebenklassen,
[13,7,1] und [5,11,17]
Alle Nebenklassen haben gleichviele Elemente, Anzahl ord(a)
Darum teilt die Elementordnung die Gruppenordnung.

Eulerscher Satz für Gruppen: Die Elemenordnung teilt die Gruppenordnung

(%i94) primep(250348280129);
(%o94) true
```

```
(%i95) pmod(2,340,341); factor(341);
(%o95) 1
(%o96) 11 31
```

4 Betrachtungen für n=p q

4.1 Definitionen

4.2 Tafeln

```
(%i104) tafel(3,5);
```

Zahlen-Tafel bis 3 *mal* 5

(%o104)

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

```
(%i105) p:3;q:5;tafel(p,q);modptafel(p,q); modqtafel(p,q);
(%o105) 3
(%o106) 5
```

Zahlen-Tafel bis 3 *mal* 5

(%o107)

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

ZahlenTafel modulo 3

(%o108)

1	2	0	1	2
0	1	2	0	1
2	0	1	2	0

ZahlenTafel modulo 5

(%o109)

1	2	3	4	0
1	2	3	4	0
1	2	3	4	0

```
(%i110) p:5;q:7;tafel(p,q);modptafel(p,q); modqtafel(p,q);
(%o110) 5
(%o111) 7
```

Zahlen-Tafel bis 5 *mal* 7

(%o112)

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35

ZahlenTafel modulo 5

(%o113)

1	2	3	4	0	1	2
3	4	0	1	2	3	4
0	1	2	3	4	0	1
2	3	4	0	1	2	3
4	0	1	2	3	4	0

ZahlenTafel modulo 7

(%o114)

1	2	3	4	5	6	0
1	2	3	4	5	6	0
1	2	3	4	5	6	0
1	2	3	4	5	6	0
1	2	3	4	5	6	0

(%i115) p:11;q:13;tafel(p,q);modptafel(p,q); modqtafel(p,q);

(%o115) 11

(%o116) 13

Zahlen-Tafel bis 11 *mal* 13

(%o117)

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140	141	142	143

ZahlenTafel modulo 11

(%o118)

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

ZahlenTafel modulo 13

(%o119)

1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0
1	2	3	4	5	6	7	8	9	10	11	12	0