KONGRUENZRELATIONEN, RESTKLASSEN, MODULARE ARITHMETIK

Wir haben auf $\mathbb{Z}$ eine Äquivalenzrelation

$$x \equiv y \mod n \quad , \quad x, y \in \mathbb{Z}, \; n \in \mathbb{N}^+$$

eingeführt, die die Menge $\mathbb{Z}$ in n Restklassen

$$[x] = \{ y \in \mathbb{Z} \mid y = k \cdot n + x \; , \; k \in \mathbb{Z} \}$$

einteilt, bzw. partitioniert. Die Menge der dadurch entstehenden Restklassen haben wir mit

$$\mathbb{Z}_n = \underbrace{\{0, \ldots, n-1\}}_{n}$$

bezeichnet.

Definiere auf $\mathbb{Z}_n$ Addition und Multiplikation durch

$$\oplus_n : \mathbb{Z}_n \times \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$$

$$z = (x+y) \mod n$$

Addit. mod n

und

$$\otimes_n \; \mathbb{Z}_n \times \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$$

$$z = (x \cdot y) \mod n \, .$$

$\boxed{n = 5}$ $\quad \oplus_5 \quad x + y \mod 5 \quad \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

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

Zu jedem $x \in \mathbb{Z}_5$ $\exists (-x) \in \mathbb{Z}_5$ $x + (-x) = 0$

$\otimes_5$ $\quad x \cdot y \mod 5$

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

$\forall x \in \mathbb{Z}_5 \setminus \{0\}$

$\exists x^{-1} \in \mathbb{Z}_5 \setminus \{0\}$

und $x \cdot x^{-1} \equiv 1 \mod 5$

$\Rightarrow (\mathbb{Z}_5, \oplus_5, \otimes_5)$ ist ein Körper.

<u>n = 2</u> $\qquad \mathbb{Z}_2 = \{0, 1\}$

$\oplus_2$	0	1
0	0	1
1	1	0

XOR - Funktion

$\otimes_2$	0	1
0	0	0
1	0	1

AND - Funktion.

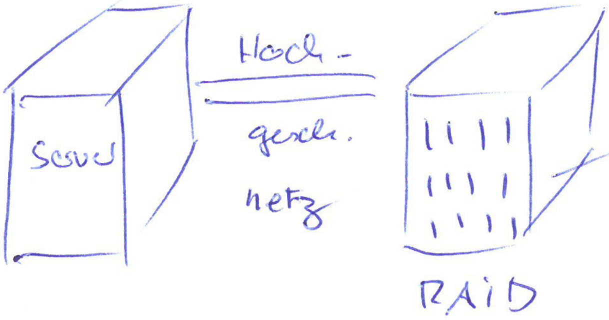
Da $\quad 0 + 0 = 0$

$\quad\quad 1 + 1 = 0 \quad$ in $\mathbb{Z}_2$ $\Rightarrow \quad a \oplus a = 0$

d.h. Addition ist das Gleiche wie Subtraktion.

Intermezzo: RAID-Systeme (David Patterson ~ 1980)

RAID = Redundant Array of Independent Disks

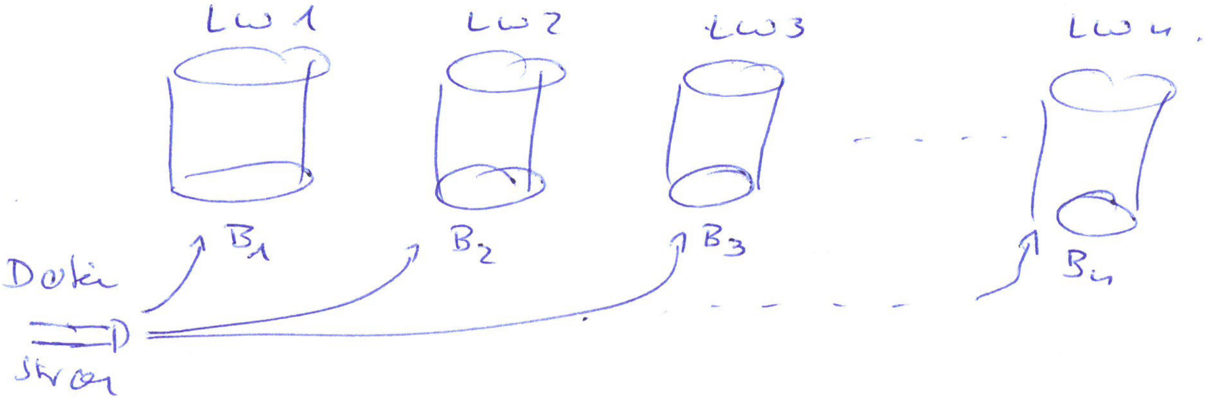


RAID-Controller
(2) = Prozessor.

Sinn & Zweck: – Erhöhung der Zugriffsgeschwindigkeit
– " " Ausfallsicherheit.

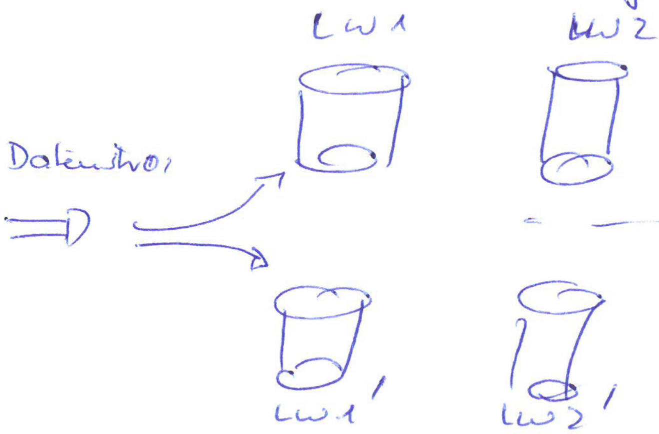
Es gibt verschiedene Levels, L0 bis L5.

Level 0 steht für reines Striping.



→ Hohe Chancen für Performance-Steigerung
→ Ausfallsicherheit nicht vorhanden.

Level 1: Mirroring (Spiegelung)



→ Sehr hohe Ausfallsicherheit

→ keine Performance-Steigerung

→ Ineffektive Speicherplatz-Nutzung.

In der Praxis: Level 0 + Level 1 10/01

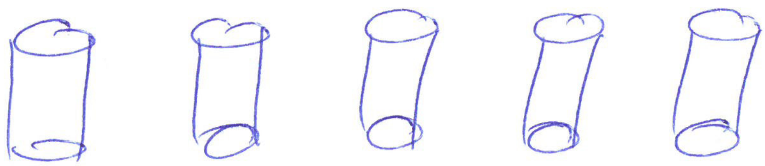
Level 2, 3 und 4 werden in der Praxis nicht genutzt,

Level 2 komplizierter Hamming-Code

Level 3 Bitweises Striping der Daten.

Level 4 sofort

Level 5



→ Daten-Strom	B_1	B_2	B_3	B_4	P_1 ($B_1 - B_4$)
	B_5	B_6	B_7	P_2(5–8)	B_8
	B_9	B_{10}	P_3	B_{11}	B_{12}

XOR

B_1	100110	
B_2	011101	Nutzdaten
B_3	100111	
B_4	011100	

P_1 = 000000

LW2 geht ins digitale Nirwana †

Der RAID-Controller rekonstruiert im laufenden Betrieb die fehlenden Datenblöcke auf eine Ersatzplatte („Hot Spare").

Der RAID-Controller berechnet die Parität der noch vorhandenen Datenblöcke

B_1	100 110
B_3	100 111
B_4	011 100
P_{neu}	011 101

Anschließend: Syndrom $P_{alt} \oplus P_{neu}$

P_{alt}	000 000
P_{neu}	011 101
B_2	011 101

Das funktioniert, weil:

$$P_{neu} \oplus P_{alt} = (B_1 \oplus B_3 \oplus B_4) \oplus (B_1 \oplus B_2 \oplus B_3 \oplus B_4)$$

$$= (B_1 \oplus B_1) \oplus B_2 \oplus (B_3 \oplus B_3) \oplus (B_4 \oplus B_4)$$

$$= 0 \oplus B_2 \oplus 0 \oplus 0 = \underline{\underline{B_2}}$$

<u>$n = 9$</u> Berechne $x \cdot y \mod 9$ | $\mathbb{Z}_9 = \{0, \ldots, 8\}$

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

Problem: Es gibt Zahlen (3 und 6) mit

$a \circ b = 0$, $a \neq 0, b \neq 0$.

oder: <u>Nicht</u> für jedes $x \in \mathbb{Z}_9 \setminus \{0\}$ gibt es ein x^{-1} mit

$x \cdot x^{-1} \equiv 1 \mod 9$.

d.h. man findet nicht für jedes $x \in \mathbb{Z}_9 \setminus \{0\}$ ein multiplikatives Inverses, d.h. <u>Division ist im allg. nicht möglich!</u>

Es klappt für $x = 1, 2, 4, 5, 7, 8$

es klappt nicht für $x = 3, 6$.

Kriterium: x^{-1} existiert, falls $\text{ggT}(x, 9) = 1$.

Rechnen mit Arithmetik mod n

Satz: Sei $a \equiv b \mod n$

und $c \equiv d \mod n$

Dann ist

$$(a \pm c) \mod n \equiv (b \pm d) \mod n$$

$$(a \cdot c) \mod n \equiv (b \cdot d) \mod n$$

1460
1460
438
3358

Beispiel:

$$146 \equiv 11 \mod 15$$

$$23 \equiv 8 \mod 15$$

$$\left(\begin{array}{l} a \equiv b \mod n \\ c \equiv d \mod n \end{array} \right)$$

$$146 \cdot 23 = 3358 \mod 15 \equiv 13$$

einfacher: $11 \cdot 8 = 88 \mod 15 \equiv 13$

Oder in einer etwas anderen Form:

$$\boxed{\begin{array}{l} (a \pm b) \mod n \equiv [(a \mod n) \pm (b \mod n)] \mod n \\ (a \cdot b) \mod n \equiv [(a \mod n) \cdot (b \mod n)] \mod n \end{array}}$$

z.B.

$$(15 \cdot 17 \cdot 23) \mod 8$$

$$\equiv [(15 \mod 8)(17 \mod 8)(23 \mod 8)] \mod 8$$

$$= (7 \cdot 1 \cdot 7) \mod 8 \equiv \underline{\underline{1 \mod 8}}$$

Division:

Satz (verallg. des obigen Beispiels $\mathbb{Z}_9$)

Seien $a, b, c \in \mathbb{Z}$ und $n \in \mathbb{N}_+$ und $ggT(a,n) = 1$

$\Rightarrow$ Wenn $(a \cdot b) \equiv (a \cdot c) \bmod n$

$\Rightarrow$ $b \equiv c \bmod n$ (kürzen / dividieren durch a nur wenn $ggT(a,n) = 1$)

Beispiel: Man löse die Gleichung

$$2x + 7 \equiv 3 \bmod 17$$

$$\Leftrightarrow \quad 2x \equiv (3-7) \bmod 17$$

$$\equiv -4 \bmod 17$$

$$\equiv 13 \bmod 17$$

$$\Leftrightarrow \quad 2x \equiv 13 \bmod 17$$

da $ggT(17,2) = 1$

$$\Leftrightarrow \quad x = (2^{-1} \cdot 13) \bmod 17$$

	0	…	2^{-1}	…	16
0					
2	…	…	1		
16					

Wie findet man $2^{-1} \bmod 17$

① Raten
② Tabelle erstellen
③ Ausrechnen.

wegen ① und der Überlegung $2 \cdot 9 = 18 \equiv 1 \bmod 17$

$2^{-1} \bmod 17 \equiv 9$

$$\Rightarrow \quad x = (2^{-1} \cdot 13) \bmod 17$$
$$= (9 \cdot 13) \bmod 17$$
$$= 117 \bmod 17$$
$$\underline{\underline{\equiv 15 \bmod 17}}$$

Check: $2x + 7 \equiv 3 \bmod 17$

$x = 15 \rightarrow 37 \equiv 3 \bmod 17$ ✓

Kleine Übung: Lösen Sie

$$5x + 6 \equiv 13 \bmod 11.$$

$$5x \equiv 7 \bmod 11$$

$ggT(5, 11) = 1$

$$x \equiv (5^{-1} \cdot 7) \bmod 11$$

$$5^{-1} \cdot 5 \equiv 1 \bmod 11$$

oder: $5^{-1} \cdot 5 =$ Vielfaches von $11 + 1$

$= k \cdot 11 + 1$

$5^{-1} = 9$, da $9 \cdot 5 = 45 = 4 \cdot 11 + 1$

$\rightarrow \quad 5^{-1} \equiv 9 \mod 11$

damit: $\quad x = (9 \cdot 7) \mod 11$

$\equiv 63 \mod 11$

$\equiv \underline{\underline{8 \mod 11}}$

Check: $\quad 5x + 6 \equiv 13 \mod 11$

$\equiv 2 \mod 11$

und $\quad 5 \cdot 8 + 6 = 46 \equiv 2 \mod 11$

Wozu braucht man das alles? Motivation: RSA-Alg.

(1) Man wähle zwei zufällige, große Primzahlen p, q, je 512, 1024, 2048 oder 4096 Bit, berechne $n = p \cdot q$.

(2) Man berechne

$$\phi(n) = (p-1)(q-1)$$

(3) Wähle eine Zahl e mit $1 < e < \phi(n)$ und $ggT(e, \phi(n)) = 1$.

(4) Berechne d mit

$$e \cdot d \equiv 1 \mod \phi(n)$$

(5) $\boxed{K_{pub} = (e, n), \quad K_{priv} = (d, n)}$

Verschlüsselung: Sei M der Klartext, dieser wird geeignet durch Zahlen codiert, z.B. ASCII-Werte.

BA → 6665

$$M < n$$

$$C = M^{e} \bmod n \qquad K_{pub} = (e, n)$$

Entschlüsselung:

$$M = C^{d} \bmod n \qquad K_{priv} = (d, n)$$

Modulare Exponentiation

oder: Wie berechnet man Monster der Form

$$z = 2^{1234} \bmod 789 \quad ?$$

Man zerlegt den Exponenten in eine Summe von 2er-Potenzen.

$$1234 = 1024 + 128 + 64 + 16 + 2$$

$$2^{1234} \bmod 789 = 2^{1024+128+64+16+2} \bmod 789$$

$$a^{b+c} = a^b \cdot a^c$$

$$= \left(2^{1024} \cdot 2^{128} \cdot 2^{64} \cdot 2^{16} \cdot 2^{2}\right) \bmod 789$$

$$(a \cdot b) \bmod n \equiv [(a \bmod n \cdot b \bmod n)] \bmod n$$

$$= \left[(2^{1024} \bmod 789)(2^{128} \bmod 789)(2^{64} \bmod 789)(2^{16} \bmod 789) \cdot (2^{2} \bmod 789)\right] \bmod 789$$

Nebenrechnung

$$2^2 \bmod 789 = 4 \bmod 789$$

$$2^4 \bmod 789 = 16 \bmod 789$$

$$2^8 \quad " \quad = 256 \bmod 789$$

$5^{28} \mod 11$

$28 = 16 + 8 + 4$

$5^{2} \mod 11 = 3$

$5^{4} \mod 11 = 9$

$5^{8} \mod 11 = 4$

$5^{16} \mod 11 = 5$

$$5^{28} \mod 11 = (\cancel{\text{[illegible]}} \cdot 9 \cdot 4 \cdot 5) \mod 11$$

$$= \underline{\underline{4 \mod 11}}$$

$$2^{16} \bmod 789 = 2^{8+8} \bmod 789$$

$$= 2^8 \cdot 2^8 \bmod 789$$

$(a \cdot b) \bmod = (a \bmod \cdot b \bmod)_n$ $$\equiv \left[\left(2^8 \bmod 789\right)\left(2^8 \bmod 789\right)\right] \bmod 789$$

$$= 256^2 \bmod 789$$

$$\equiv 49 \bmod 789$$

$$2^{32} \bmod 789 \equiv 49^2 \bmod 789 \equiv 34 \bmod 789$$

$$2^{64} \bmod 789 \equiv 34^2 \bmod 789 \equiv 367 \bmod 789$$

$34 \cdot 34 = 1156$ $\qquad$ $1156 : 789 = 1{,}465\ldots\ldots$

$0{,}465\ldots * 789 = 367$

$$2^{128} \bmod 789 \equiv 367^2 \bmod 789 = 559$$

$$\left.\begin{aligned} 2^{256} \bmod 789 &= 37 \\ 2^{512} &= 580 \\ 2^{1024} &= 286 \end{aligned}\right\} \bmod 789$$

$$\Rightarrow 2^{1234} \bmod 789 = \left[286 \cdot 559 \cdot 367 \cdot 49 \cdot 4\right] \bmod 789$$

$$\equiv \underline{\underline{481 \bmod 789}}$$

Übung! Berechnen Sie

$$5^{28} \bmod 11.$$

$$28 = 16 + 8 + 4$$

$\Rightarrow 5^{28} \bmod 11 = (5^{16} \cdot 5^{8} \cdot 5^{4}) \bmod 11$

$\equiv \left[(5^{16} \bmod 11)(5^{8} \bmod 11) \cdot (5^{4} \bmod 11)\right] \bmod 11$

$5^2 \bmod 11 = 25 \bmod 11 \equiv 3$

$5^4 \bmod 11 \equiv 3^2 \bmod 11 \equiv 9$

$5^8 \bmod 11 \equiv 81 \bmod 11 \equiv 4$

$5^{16} \bmod 11 \equiv 4^2 \bmod 11 \equiv 16 \bmod 11 \equiv 5$

$\equiv (5 \cdot 4 \cdot 9) \bmod 11$

$\equiv (9 \cdot 9) \bmod 11 \equiv \underline{4 \bmod 11}$

Algorithmus Modulare Exponentiation

Der folgende Alg. berechnet fastexp $= x^y \bmod n$,

Input: Drei positive ganze Zahlen x, y, n.
Drei Hilfsvariable a, b, c.

Initial. $a \leftarrow x$,
$b \leftarrow 1$,
$c \leftarrow y$

(*) Falls c gerade, setze
$a \leftarrow a^2 \bmod n$
$b \leftarrow b$
$c \leftarrow \frac{c}{2}$

Falls c ungerade, setze
$a \leftarrow a$
$b \leftarrow (a \cdot b) \bmod n$
$c \leftarrow c - 1$

Falls $c \neq 0$ weiter mit (*), sonst

STOP, output: $b = x^y \bmod n$

Bsp: $5^{28} \bmod 11$

$x = 5$, $y = 28$, $n = 11$

$a \leftarrow 5$

$b \leftarrow 1$

$c \leftarrow 28$

1. Runde: c gerade

$a \leftarrow a^2 \bmod n = 5^2 \bmod 11 = 3$

$b \leftarrow b = 1$

$c \leftarrow 14$

2. Runde: c gerade

$a \leftarrow a^2 \bmod n = \overset{9}{\cancel{16}} \bmod 11 = 9$

$b \leftarrow 1$

$c \leftarrow 7$

3. Runde: c ungerade

$a \leftarrow a = 9$

$b \leftarrow (a \cdot b) \bmod n = \cancel{4}\,9$

$c \leftarrow c - 1 = 6$

4. Runde: c gerade

$a \leftarrow a^2 \bmod n = \overset{9^2}{\cancel{25}} \bmod 11 = \cancel{6}\,4$

$b \leftarrow b = \cancel{4}\,9$

$c \leftarrow \frac{c}{2} = 3$

5. Runde: c ungerade

$a \leftarrow 4$, $b \leftarrow a \cdot b \bmod n = \cancel{4}\,3$

$c \leftarrow c - 1 = 2$

6. Runde $\quad c$ gerade

$$a \leftarrow a^2 \bmod n = 4^2 \bmod 11 = 5$$

$$b \leftarrow b = \cancel{4}\,3$$

$$c \leftarrow 1$$

7. Runde $\quad c$ ungerade

$$a \leftarrow a = \cancel{9}\,5$$

$$b \leftarrow (a \cdot b) \bmod n = \cancel{36}^{\,15} \bmod 11 = \cancel{3}\,4$$

$$c \leftarrow 0$$

$$\Rightarrow \quad 5^{28} \bmod 11 \equiv \cancel{3}\,\underline{\underline{4}}$$

Das systematische Verfahren, um das multiplikative Inverse modulo n zu berechnen, ist der erweiterte Euklidische Algorithmus.

<u>Beispiel</u>: Berechne das multiplikative Inverse von 510 modulo 1001, d.h. gesucht ist die Zahl $x \in \mathbb{Z}_{1001}$ mit

$$510 \cdot x \equiv 1 \bmod 1001.$$

(Divisionsverfahren).

<u>1. Dividiere</u> 1001 durch 510

$$1001 = 1 \cdot 510 + 491.$$

In jedem Schritt: Der Rest der Division wird ausgedrückt als Linearkomb. von 510 und 1001.

$$491 = 1 \cdot 1001 - 1 \cdot 510.$$

2. Dividiere 510 durch 491

$$510 = 1 \cdot 491 + 19$$

$$19 = 510 - 491$$
$$= 510 - 1 \cdot 1001 + 1 \cdot 510$$
$$= 2 \cdot \underline{510} - 1 \cdot \underline{1001}$$

3. Dividiere 491 durch 19

$$491 = 25 \cdot 19 + 16$$

$$16 = 491 - 25 \cdot 19$$
$$= 1 \cdot 1001 - 1 \cdot 510 - 25(2 \cdot 510 - 1 \cdot 1001)$$
$$= \underline{26 \cdot 1001 - 51 \cdot 510}$$

4. Dividiere 19 durch 16

$$19 = 1 \cdot 16 + 3$$

$$3 = 19 - 16 = 2 \cdot 510 - 1 \cdot 1001$$
$$- 26 \cdot 1001 + 51 \cdot 510$$
$$= 53 \cdot 510 - 27 \cdot 1001$$

5. Dividiere 16 durch 3

$$16 = 5 \cdot 3 + 1$$

$$1 = 16 - 5 \cdot 3 = 26 \cdot 1001 - 51 \cdot 510$$
$$-5(53 \cdot 510 - 27 \cdot 1001)$$

STOP:

$$1 = 161 \cdot 1001 - 316 \cdot 510$$

$$\Rightarrow -316 \cdot 510 = -161 \cdot 1001 + 1$$

$$-316 \cdot 510 \equiv 1 \mod 1001$$

$$\Rightarrow 510^{-1} \mod 1001 = -316 \mod 1001$$

$$\equiv 685 \mod 1001$$

Check: Es gilt:

$$510 \cdot 685 = 349350$$

$$= 349349 + 1$$

$$= 349 \cdot 1001 + 1$$

$$\equiv 1 \mod 1001.$$

Algorithmus: Erweiterter EUKLID

Input: Zwei positive Zahlen a, b, o.B.d.A.: $a < b$

Hilfsvariable $X_1 \quad X_2 \quad X_3$

$Y_1 \quad Y_2 \quad Y_3$

$T_1 \quad T_2 \quad T_3$, Q.

Initial. $X_1 \leftarrow 1 \qquad Y_1 \leftarrow 0$

$X_2 \leftarrow 0 \qquad Y_2 \leftarrow 1$

$X_3 \leftarrow \underline{\underline{b}} \qquad Y_3 \leftarrow a$

(*) Falls $Y_3 = 0$ STOP. return $X_3 = ggT(a,b)$

kein Inverses

Falls $Y_3 = 1$ STOP return $Y_2 = a^{-1} \bmod b$.

Sonst: $Q = \left\lfloor \frac{X_3}{Y_3} \right\rfloor$

Berechne $T_1 = X_1 - QY_1$

$T_2 = X_2 - QY_2$

$T_3 = X_3 - QY_3$

Setze: $X_1 \leftarrow Y_1 \qquad Y_1 \leftarrow T_1$

$X_2 \leftarrow Y_2 \qquad Y_2 \leftarrow T_2$

$X_3 \leftarrow Y_3 \qquad Y_3 \leftarrow T_3$

weiter mit (*)

T_1 : Vielfache von 1001

T_2 " " 510