

Faktorenzerlegung großer Zahlen

3.) Faktorenzerlegung à la Monte Carlo :

Bei der Methode von J. M. Pollard ist es nicht immer möglich bei einer teilbaren Zahl einen Faktor zu finden. Die Suche wird zu einem Glücksspiel. Man untersucht ob es von einer natürlichen Zahl **N** und einer errechneten Zahl **P** einen größten gemeinsamen Teiler gibt. **P** bekommt man, indem zwei neue Variablen (**x**, **y**) einführt, für welche gilt : $f(x)=x^2+c$ ($c=0;2$) $x \leftarrow x^2+c$; $y \leftarrow (y^2+c)^2+c$ »»»» $P_2=P_1 \cdot (y - x)$

Als Startwert für **x**, **y**, und **P** wird **1** verwendet.

1.Schritt:

Setze **x=1 ;y=1 ;P=1**

2.Schritt:

Setze $x = x^2+c$; $y = (y^2+c)^2+c$ und $P_2=P_1 \cdot (y - x)$

3.Schritt:

Berechne den **ggT** von **P** und **N**. Falls das Ergebnis = **1** ist , Wiederhole den Vorgang ab dem Schritt 2 . Bei allen anderen Ergebnissen hat man mit dem **ggT** den gesuchten Teiler gefunden .

!Achtung!:

- Bei Schritt 2 entstehen für die Variablen **x**, **y** und **P** große Werte. Man muß daher die Ergebnisse modulo **N** nehmen .
- Falls **N** selbst der Teiler ist, gibt es zwei Möglichkeiten:
 1. **N** ist eine Primzahl
 2. Man kann die Konstante **C** verändern (z.B.:von 1 auf 2,3,...)
- Wenn einem die Berechnung des Schrittes 3 zu Zeitaufwendig ist, kann man dies umgehen, indem man erst nach z.B.: 10 -maliger Wiederholung des 2. Schrittes mit dem 3.Schritt beginnt .(Nur sehr seltener Verlust des Teilers) .

Hier ein Programmierbeispiel in Q-Basic :

```

1. CLS: CLEAR: X=1: Y=1: P=1: T=1 ' **** Programm zur Faktorenzerlegung von MEISEL Marcus ****
2. INPUT "zu teilende Zahl (N): "; N: MO=N: INPUT "Konstante C: "; C: CLS
3. X=X^2+C: GOSUB 9: Y=(Y^2+C)^2+C: GOSUB 10: P=P*(Y-X): GOSUB 11: V=V+1: IF V>=25 THEN V=25
4. LOCATE 10,0: PRINT "X      Y      P": LOCATE 10,V: PRINT X:; LOCATE 23,V: PRINT Y:;
   LOCATE 36,V: PRINT P;
5. IF P=0 THEN GOTO 19
6. IF P=1 THEN GOTO 3
7. INPUT "": I: GOTO 12
8. GOTO 3
9. X=X-(INT(X/MO))*MO: RETURN
10. Y=Y-(INT(Y/MO))*MO: RETURN
11. IF ABS ( P ) <> P THEN P= N - ABS ( P ) : P=P - ( INT ( P / MO ) ) * MO : RETURN
12. ' *****ggT --ausrechnen
13. TE = P
14. RE = N - ( INT ( N / TE ) ) * TE: IF RE=1 THEN T=1: GOTO 3
15. IF RE=0 THEN T=TE: GOTO 17
16. N=TE: TE=RE: GOTO 14
17. ' *****die Lösung
18. CLS: LOCATE 1,1: PRINT "Zahl "; MO; " u. "; P; " sind durch "; T; " Teilbar.": BEEP: INPUT "" : I: GOTO 1
19. ' *****eine Primzahl

```

```
20. CLS:LOCATE 1,1:PRINT "Zahl ";MO;"ist eine Primzahl! -oder ein anderes !!c!! prob.";BEEP:INPUT"" ;I
21. GOTO1          ' ***** ENDE dieses ©Programs
Hier ein RechenBeispiel:
```

N=143
c=1

=>	X	Y	P
	1	1	1
	2	5	3
	5	105	14
	26	83	83
	105	105	0=> Primzahl oder c anders wählen z.B.: c=2

N=143
c=2

=>	X	Y	P
	3	11	8
	11	116	125
	123	115	142
	116	38	65 → ggT(65,143)=13 => <u>13/143</u>

Antwort : 143 ist durch 13 teilbar.

4.) Weitere Algorithmen :

- Ein weiterer Algorithmus von J. M. Pollard:
Es wird ein Teiler **p** von einer Zahl **N** gesucht. Er ist gefunden, wenn gilt : **p-1** ist nur durch kleine Primfaktoren teilbar. Auf diesen Rechengang wird in dem Artikel von Williams näher eingegangen.
- Der SQUFOF-Algorithmus von D. Shanks:
Hierfür benötigt man die Theorie der Ideale in quadratischen Zahlkörpern. Näheres : siehe Artikel von Williams.