

```

[ > restart:
Wählen Primzahlen p und q, berechnen N und phi(N):
> p := 13:
  q := 11:
  N := p*q;
  phi_N := (p-1)*(q-1);

                                     N := 143
                                     phi_N := 120

Wähle einen Exponenten e:
> e := 23;

                                     e := 23

Teste, ob e teilerfremd zu phi(N) ist und berechne gleichzeitig s und t mit  $s \cdot \phi(N) + t \cdot e = \text{ggT}(\phi(N), e)$ 
> igcdex(phi_N, e, 's', 't');
  s, t;

                                     1
                                     -9, 47

Für  $d := t$  gilt also  $d \cdot e \bmod \phi(N) = \text{ggT}(\phi(N), e) = 1$ 
> d := t mod phi_N;
  d * e mod phi_N;

                                     d := 47
                                     1

Wähle eine Nachricht x:
> x := 15;

                                     x := 15

 $x^e$  ist schon ganz schön groß, aber mod N ist es in  $\{0, \dots, 142\}$ :
> x^e;
  x^e mod N;

                                     1122274146401882171630859375
                                     20

Rekursives, modulares Potenzieren
> poti := proc(b :: integer, e :: integer, n :: posint)
  if e = 0 then return 1;
  elif e mod 2 = 0 then return procname( b, e/2, n )^2 mod n;
  else return b * procname( b, e-1, n ) mod n;
  end if;
end proc:

Verschlüsseln:
> y := poti( x, e, N );

                                     y := 20

Entschlüsseln:
> poti( y, d, N );

                                     15

Maple-interne Routinen:
> x ^ e mod N;
  % ^ d mod N;

                                     20
                                     15

```

Die Maple Routine braucht schon bei ziemlich kleinen Zahlen recht lang. Das liegt daran, weil erst am Schluss eine Division mit Rest ausgeführt wird. Wenn Ihr das  $\wedge$  durch ein  $\&\wedge$  ersetzt, wird nach jedem Schritt "reduziert" (also Division mit Rest durch 1392134 durchgeführt) und das ganze geht auch für sehr große Zahlen sehr schnell.

```
> st := time():  
  120472 ^ 23042 mod 1392134;  
  time() - st;  
  
                                     911532  
                                     0.681
```

[ >