

```

> pptest := proc (N::integer,t::integer)
#
# Probabilistischer Primzahltest, der mit
# zufällig gewählten Startwerten  $1 < a < N$ 
# maximal t Iterationen von
#   Teilbarkeitstest,
#   ggT-Test,
#   Fermat-Kongruenztest
#   Miller-Rabin-Test
# durchführt.
#
local a,d,e,j,k,s,u,f,found,randomelement;
randomelement := rand(2..N-1);
for s from 1 to t do
a := randomelement();
#
# Teilbarkeitstest
#
if N mod a = 0 then
RETURN(N,`ist keine Primzahl:`,a,`teilt`,N) fi;
#
# Euklid-Test
#
d := igcd(a,N);
if d>1 then
RETURN(N,`ist keine Primzahl: der GGT von`,a,`und`,N,`ist`,d) fi;
#
# Fermat-Test
#
e := a &^(N-1) mod N;
if not (e=1) then
RETURN(N,`ist keine Primzahl:
Fermat-Test mit`,a,`liefert  $a^{(N-1)} \bmod N =`,e) fi;
#
# Miller-Rabin-Test
#
print(`Miller-Rabin-Test wird gestartet`);
u := N-1;
k := 0;
while (u mod 2 = 0) do
    u := u/2;
    k := k+1;
od;
f := a &^u mod N;
print(f);
if (f=-1 mod N) or (f=1 mod N) then break fi;
found := false;
for j from 1 to k do
    f := f &^2 mod N;
    print(f);
    if f=-1 mod N then found := true; break fi;
od;
if found then break else
RETURN(N,`ist keine Primzahl: Miller-Rabin-Test!`)$ 
```

```
fi;  
od;  
print(`MR-Test`,t,`-mal bestanden:`,N,` ist vermutlich Primzahl`);  
1;  
end;
```

```
> pptest(41,3);
```

Miller-Rabin-Test wird gestartet

1

MR-Test, 3, -mal bestanden.: 41, ist vermutlich Primzahl

1

```
> pptest(561,5);
```

Miller-Rabin-Test wird gestartet

287

463

67

1

1

561, ist keine Primzahl: Miller-Rabin-Test!

```
> pptest(10007,5);
```

Miller-Rabin-Test wird gestartet

10006

MR-Test, 5, -mal bestanden.: 10007, ist vermutlich Primzahl

1

```
> isprime(10007);
```

true

```
> findprime := proc (N::integer,t::integer)  
#  
# die nächstgrössere Primzahl nach N  
# wird mittels probabilistische Primzahltest  
# bestimmt  
# t = Anzahl der Iterationen  
#  
local n;  
n := N;  
if (n mod 2 = 0) then n := n+1 fi;  
while true do  
if pptest(n,t)=1 then break fi;  
n := n+2;  
od;  
end;
```

findprime := proc(N::integer, t::integer)

local n;

n := N;

if `mod` (n, 2) = 0 then n := n + 1 end if

```
do if  $pp_{test}(n, t) = 1$  then break end if  $n := n + 2$  end do  
end proc
```

```
> findprime(8234857628934756343, 5);
```

```
Miller-Rabin-Test wird gestartet
```

```
7156348946777158704
```

```
3903333351366841698
```

```
8234857628934756360
```

```
MR-Test, 5, -mal bestanden.: 8234857628934756361, ist vermutlich Primzahl
```

```
8234857628934756361
```

```
> isprime(%);
```

```
true
```

```
>
```