

Primzahltests in MuPAD

- `isprime1:=proc(n) local a;`
`begin`
`if n < 2 then return(FALSE); end_if;`
`for a from 2 to n-1 do`
`if igcd(a,n) > 1 then return(FALSE); end_if;`
`end_for;`
`return(TRUE);`
`end_proc:`
- `isprime2:=proc(n) local a;`
`begin`
`if n < 2 then return(FALSE); end_if;`
`for a from 2 to trunc(sqrt(n)) do`
`if igcd(a,n) > 1 then return(FALSE); end_if;`
`end_for;`
`return(TRUE);`
`end_proc:`
- `isprime3:=proc(n) local a;`
`begin`
`if n<2 then return(FALSE); end_if;`
`for a from 2 to n-1 do`
`if igcd(a, n) = 1 and a^(n-1) mod n <> 1 then return(FALSE); end_if;`
`end_for;`
`return(TRUE);`
`end_proc:`
- `isprime4:=proc(n)`
`begin`
`if n<2 then return(FALSE); end_if;`
`if n=2 or n=3 or n=5 then return(TRUE); end_if;`
`if igcd(2, n) <> 1 or 2^(n-1) mod n <> 1 then return(FALSE); end_if;`
`if igcd(3, n) <> 1 or 3^(n-1) mod n <> 1 then return(FALSE); end_if;`
`if igcd(5, n) <> 1 or 5^(n-1) mod n <> 1 then return(FALSE); end_if;`
`return(TRUE);`
`end_proc:`
- `isprime4powermod:=proc(n)`
`begin`
`if n<2 then return(FALSE); end_if;`
`if n=2 or n=3 or n=5 then return(TRUE); end_if;`
`if igcd(2, n) <> 1 or powermod(2,n-1,n) <> 1 then return(FALSE); end_if;`
`if igcd(3, n) <> 1 or powermod(3,n-1,n) <> 1 then return(FALSE); end_if;`
`if igcd(5, n) <> 1 or powermod(5,n-1,n) <> 1 then return(FALSE); end_if;`
`return(TRUE);`
`end_proc:`
- `isprime?Liste:=proc(n) local i, Liste;`
`begin`
`Liste:=[];`
`for i from 1 to n do`
`if isprime?(i)=TRUE then Liste:=append(Liste, i); end_if;`
`end_for;`
`return(Liste, nops(Liste));`
`end_proc:`
- `isprime5:=proc(n)`
`begin`
`if n<2 then return(FALSE); end_if;`
`if n=2 or n=3 or n=5 then return(TRUE); end_if;`
`if igcd(2, n) <> 1 or igcd(3, n) <> 1 or igcd(5, n) <> 1 then return(FALSE); end_if;`
`if powermod(2,(n-1)/2,n) <> 1 and powermod(2,(n-1)/2,n) <> n-1 then return(FALSE); end_if;`
`if powermod(3,(n-1)/2,n) <> 1 and powermod(3,(n-1)/2,n) <> n-1 then return(FALSE); end_if;`
`if powermod(5,(n-1)/2,n) <> 1 and powermod(5,(n-1)/2,n) <> n-1 then return(FALSE); end_if;`
`return(TRUE);`
`end_proc:`

Primzahltests in MuPAD

- `isprime1:=proc(n) local a;
begin
 if n < 2 then return(FALSE); end_if;
 for a from 2 to n-1 do
 if igcd(a,n) > 1 then return(FALSE); end_if;
 end_for;
 return(TRUE);
end_proc:`
- `isprime2:=proc(n) local a;
begin
 if n < 2 then return(FALSE); end_if;
 for a from 2 to trunc(sqrt(n)) do
 if igcd(a,n) > 1 then return(FALSE); end_if;
 end_for;
 return(TRUE);
end_proc:`
- `isprime3:=proc(n) local a;
begin
 if n<2 then return(FALSE); end_if;
 for a from 2 to n-1 do
 if igcd(a, n) = 1 and a^(n-1) mod n <> 1 then
 return(FALSE); end_if;
 end_for;
 return(TRUE);
end_proc:`
- `isprime4:=proc(n)
begin
 if n<2 then return(FALSE); end_if;
 if n=2 or n=3 or n=5 then return(TRUE); end_if;
 if igcd(2, n) <> 1 or 2^(n-1) mod n <> 1 then
 return(FALSE);
 end_if;
 if igcd(3, n) <> 1 or 3^(n-1) mod n <> 1 then
 return(FALSE);
 end_if;
 if igcd(5, n) <> 1 or 5^(n-1) mod n <> 1 then
 return(FALSE);
 end_if;
 return(TRUE);
end_proc:`

- `isprime4powermod:=proc(n)`
`begin`
 `if n<2 then return(FALSE); end_if;`
 `if n=2 or n=3 or n=5 then return(TRUE); end_if;`
 `if igcd(2, n) <> 1 or powermod(2,n-1,n) <> 1 then`
 `return(FALSE);`
 `end_if;`
 `if igcd(3, n) <> 1 or powermod(3,n-1,n) <> 1 then`
 `return(FALSE);`
 `end_if;`
 `if igcd(5, n) <> 1 or powermod(5,n-1,n) <> 1 then`
 `return(FALSE);`
 `end_if;`
 `return(TRUE);`
`end_proc:`

- `isprime?Liste:=proc(n) local i, Liste;`
`begin`
 `Liste:=[];`
 `for i from 1 to n do`
 `if isprime?(i)=TRUE then Liste:=append(Liste, i); end_if;`
 `end_for;`
 `return(Liste, nops(Liste));`
`end_proc:`

- `isprime5:=proc(n)`
`begin`
 `if n<2 then return(FALSE); end_if;`
 `if n=2 or n=3 or n=5 then return(TRUE); end_if;`
 `if igcd(2, n) <> 1 or igcd(3, n) <> 1 or igcd(5, n) <> 1 then`
 `return(FALSE);`
 `end_if;`
 `if powermod(2,(n-1)/2,n) <> 1 and powermod(2,(n-1)/2,n) <> n-1 then`
 `return(FALSE);`
 `end_if;`
 `if powermod(3,(n-1)/2,n) <> 1 and powermod(3,(n-1)/2,n) <> n-1 then`
 `return(FALSE);`
 `end_if;`
 `if powermod(5,(n-1)/2,n) <> 1 and powermod(5,(n-1)/2,n) <> n-1 then`
 `return(FALSE);`
 `end_if;`
 `return(TRUE);`
`end_proc:`