

Test di primalità

- Note

- Autore

Claudio Marsan
Liceo Cantonale di Mendrisio
Via Agostino Maspoli
CH-6850 Mendrisio (Switzerland)
e-mail: claudio.marsan@liceomendrisio.ch

- Versione

Versione 2.0, 18 marzo 2003
Maple V Release 6.02 for Windows 2000

> **restart:**

> **with(numtheory):**

Warning, the protected name order has been redefined and unprotected

1. Il teorema di Fermat come test di compostezza

> **Fermat_comp_test := proc(n::posint, a::posint)**

if type(n, even) then

RETURN(`composto`);

end if;

if igcd(n,a)<>1 then

RETURN(`composto`);

end if;

if (a &^ (n-1) mod n)<>1 then

`composto`;

else

`Il test non è conclusivo`;

end if;

end;

>

Fermat_comp_test := proc(n::posint, a::posint)

if type(n, even) then RETURN(composto) end if;

if igcd(n, a) ≠ 1 then RETURN(composto) end if;

if &^(a, n - 1) mod n ≠ 1 then composto else `Il test non è conclusivo` end if

end proc

2. Il criterio di Eulero come test di compostezza

> **Euler_comp_test := proc(n::posint, a::posint)**

if type(n, even) then

RETURN(`composto`);

```

end if;
if igcd(n,a)<>1 then
  RETURN(`composto`);
end if;
if (a &^ ((n-1)/2) mod n)<>1 and (a &^ ((n-1)/2) mod n)<>(n-1)
then
  RETURN(`composto`);
end if;
if (a &^ ((n-1)/2) mod n)<>(J(a,n) mod n) then
  RETURN(`composto`);
end if;
`Il test non è conclusivo`;
end;

```

Euler_comp_test := proc(n::posint, a::posint)

```

if type(n, even) then RETURN(composto) end if;
if igcd(n, a) ≠ 1 then RETURN(composto) end if;
if `&^(a, 1 / 2*n - 1 / 2) mod n ≠ 1 and `&^(a, 1 / 2*n - 1 / 2) mod n ≠ n - 1 then
  RETURN(composto)
end if;
if `&^(a, 1 / 2*n - 1 / 2) mod n ≠ J(a, n) mod n then RETURN(composto) end if;
`Il test non è conclusivo`

```

end proc

```
> n:=294409; a:=7; Fermat_comp_test(n, a); Euler_comp_test(n, a);
```

```
      n := 294409
```

```
      a := 7
```

```
      Il test non è conclusivo
```

```
      composto
```

3. **Strong pseudoprime test** per la base a

```
> strong_pseudoprime := proc(n::posint, a::posint)
```

```
  local m,s,d,i;
```

```
  if type(n, even) then RETURN(false);
```

```
  end if;
```

```
  m := n-1;
```

```
  s := 0;
```

```
  while type(m, even) do
```

```
    s := s + 1;
```

```
    m := m/2;
```

```
  end do;
```

```
  d := (n - 1)/2^s;
```

```
  if (a &^d mod n) = 1 then
```

```
    RETURN(true);
```

```
  end if;
```

```
  for i from 0 to s-1 do
```

```

    if (a &^(d*2^i) mod n)=n-1 then
        RETURN(true);
    end if;
end do;
RETURN(false);
end;

```

strong_pseudoprime := **proc**(*n::posint*, *a::posint*)

local *m*, *s*, *d*, *i*;

if type(*n*, even) then RETURN(*false*) end if;

m := *n* - 1;

s := 0;

while type(*m*, even) do *s* := *s* + 1; *m* := 1 / 2**m* end do;

d := (*n* - 1) / 2^{*s*};

if $\&^{\&}(a, d) \bmod n = 1$ then RETURN(*true*) end if;

for *i* from 0 to *s* - 1 do if $\&^{\&}(a, d*2^i) \bmod n = n - 1$ then RETURN(*true*) end if

end do;

RETURN(*false*)

end proc

> **strong_pseudoprime**(25326001, 3);

true

4. I 13 "strong pseudoprimes" per le basi 1,3,5 l e minori di $25*10^9$

> **spsp235** := [25326001, 161304001, 960946321, 1157839381,
3215031751, 3697278427, 5764643587, 6770862367, 14386156093,
15579919981, 18459366157, 19887974881, 21276028621];

spsp235 := [25326001, 161304001, 960946321, 1157839381, 3215031751, 3697278427,
5764643587, 6770862367, 14386156093, 15579919981, 18459366157, 19887974881,
21276028621]

5. Strong pseudoprime test

> **Strong_pseudoprime_test** := **proc**(*n::posint*)

if type(*n*, even) then

RETURN($\&$ composto $\&$);

end if;

if $n > 25*10^9$ then

RETURN($\&$ test non applicabile $\&$);

end if;

if **strong_pseudoprime**(*n*, 2) = false then

RETURN($\&$ composto $\&$);

end if;

if **strong_pseudoprime**(*n*, 3) = false then

RETURN($\&$ composto $\&$);

end if;

if **strong_pseudoprime**(*n*, 5) = false then

```
    RETURN(`composto`);
end if;
if member(n, spsp235) then
    RETURN(`composto`);
else
    RETURN(`primo`);
end if;
end;
```

```
Strong_pseudoprime_test := proc(n::posint)
```

```
    if type(n, even) then RETURN(composto) end if;
    if 25000000000 < n then RETURN(`test non applicabile`) end if;
    if strong_pseudoprime(n, 2) = false then RETURN(composto) end if;
    if strong_pseudoprime(n, 3) = false then RETURN(composto) end if;
    if strong_pseudoprime(n, 5) = false then RETURN(composto) end if;
    if member(n, spsp235) then RETURN(composto) else RETURN(primo) end if
```

```
end proc
```

```
[ > Strong_pseudoprime_test(23111111111);
```

```
                test non applicabile
```

```
[ > Strong_pseudoprime_test(23111111113);
```

```
                primo
```

```
[ > isprime(23111111113);
```

```
                true
```

```
[ >
```