Test de non primalité,Test de primalité

Test de Rabin–Miller

```
test := proc(a, N)
    local m, q, u, i;
    m := 1;
    while Rem(N − 1, 2^(m + 1)) = 0 do
        m := m + 1
    end do;
    q := (N − 1) / 2^m;
    u := mod(Power(a, q), N);
    if u = 1 then
        return 1
    else
        i := 0;
        while mod(Power(u, 2^i), N) = mod(−1, N) and i < m do
            i := i + 1
        end do
    end if;
    if i = m then
        return 1
    else
        return 0
    end if
end proc;
    −1
```

```
test(3, 5); 1
```

1 (1.1)

```
test(2, 3277); 1
```

1 (1.2)

```
test(3, 9); 1
```

0 (1.3)

```
test(5, 17); 1
```

(1.4)

1 (1.4)

```
RM := proc(N, k)
    local i, u;
    i := 1;
    u := RandomTools[Generate](integer(range = 2 ..N - 1));
    while i < k and test(u, N) = 1 do
        u := RandomTools[Generate](integer(range = 2 ..N - 1));
        i := i + 1
    end do;
    if i = k then
        return N* est* surement* premier
    else
        return N* nest* pas* premier
    end if
end proc;
    -1
```

RM(15, 5); 1; RM(23, 8); 1

23 *est surement premier* (1.5)

RM(561, 7); 1

561 *nest pas premier* (1.6)

ithprime(148745); 1

1997293 (1.7)

RM(1997293, 18); 1

1997293 *est surement premier* (1.8)

RM(14976189, 15); 1

14976189 *nest pas premier* (1.9)

RM(14847864846718745 83, 13); 1

14847864846718745 83 *nest pas premier* (1.10)

isprime(14847864846718745 83); 1

false (1.11)

Test de Pocklington–Lehmer

```
test2 := proc(a, N, p)
    if mod(Power(a, N - 1), N) = mod(1, N)
     and igcd(a^((N - 1) / p) - 1, N) = 1 then
        return 1
    else
```

```
        return 0
    end if
end proc;
    -1

PL := proc(N, S, v)
    local m, u, i, j, o;
    m := nops(S);
    u := 1;
    for i by 2 to m do
        u := u * S[i] ^ S[i + 1]
    end do;
    if u < evalf(sqrt(N)) then
        return `d&#65533;composition` * insuffisante
    else
        u := RandomTools[Generate](integer(range = 2 .. N - 1));
        j := 1;
        while j <= m do
            o := 1;
            while test2(u, N, S[j]) = 0 and o < v do
                u := RandomTools[Generate](integer(range = 2 .. N - 1));
                o := o + 1
            end do;
            if o = v then
                return N * non * premier
            end if;
            j := j + 2
        end do;
        if m < j then
            return N * est * premier
        end if
    end if
end proc;
    -1
S := [2, 4, 3, 2, 5, 2]; 1
```

$$[2, 4, 3, 2, 5, 2] \tag{2.1}$$

```
*(9, 25) ^ (2, 4); 1
```

$$3600 \tag{2.2}$$

```
PL(3601, S, 25); 1
```

$$3601\ non\ premier \tag{2.3}$$

```
isprime(3601); 1
```

$$false \tag{2.4}$$

```
ithprime(35); 1
```

$$149 \tag{2.5}$$

```
S := [2, 2, 37, 1]; -1
PL(149, S, 28); 1
```

$$149\ est\ premier \tag{2.6}$$

ithprime(354); 1

2383 (2.7)

S:= [2, 1, 3, 1, 397, 1]; 1

[2, 1, 3, 1, 397, 1] (2.8)

PL(2383, S, 28); 1

2383 *est premier* (2.9)

^(43, 5) ^(23, 7) ^(31, 14); 1

37887926979387136561932279205613265574 1 (2.10)

S:= [23, 7, 31, 14, 43, 5]; 1

[23, 7, 31, 14, 43, 5] (2.11)

PL(37887926979387136561932279205613265574 2, S, 30); 1

37887926979387136561932279205613265574 2 *non premier* (2.12)

isprime(37887926979387136561932279205613265574 2); 1

false (2.13)

ithprime(458128); 1

6709663 (2.14)

ifactor(6709662); 1

$(2)\,(3)^3\,(17)\,(7309)$ (2.15)

S:= [7309, 1]; 1; *PL*(6709663, S, 15); 1

6709663 *est premier* (2.16)