```
floor( fsolve(x* (x-1) =2^24, x, 1..2^64) ); # modular<float>
floor( fsolve(x* (x-1) =2^53, x, 1..2^64) ); # modular<double>
floor( fsolve(x* (x-1) =2^31, x, 1..2^64) ); # modular<int32>
floor( fsolve(x* (x-1) =2^63, x, 1..2^64) ); # modular<int64>
```

4096

94906266

46341

3037000500 **(1)**

```
floor( fsolve( (x+1) * (x-1) =2^26, x, 1..2^64) ); # modular-balanced<float>
floor( fsolve( (x+1) * (x-1) =2^55, x, 1..2^64) ); # modular-balanced<double>
floor( fsolve( (x+1) * (x-1) =2^33, x, 1..2^64) ); # modular-balanced<int32>
floor( fsolve( (x+1) * (x-1) =2^65, x, 1..2^64) ); # modular-balanced<int64>
```

8192

189812531

92681

6074001000 **(2)**

```
min( solve( 2·x=2^16, x), floor( fsolve(x* (x-1) =2^31, x, 1..2^64) ) ) : %, prevprime(%);
    # ZpzDom<Std16>
floor( fsolve(x* (x-1) =2^31, x, 1..2^64) ) : %, prevprime(%);# ZpzDom<Std32>
floor( fsolve(x* (x-1) =2^32, x, 1..2^64) ) : %, prevprime(%);# ZpzDom<Unsigned32>
floor( fsolve(x* (x-1) =2^28, x, 1..2^64) ) : %, prevprime(%);# ZpzDom<Std32>
floor( fsolve(x* (x-1) =2^63, x, 1..2^64) ) : %, prevprime(%);# ZpzDom<Std64>
floor( fsolve( subs( B=2^16, (x-1)^2 + x·B=B^2), x, 1..2^64) ) : %-1, prevprime(%);
    # Montgommery<Std32>
```

32768, 32749

46341, 46337

65536, 65521

16384, 16381

3037000500, 3037000493

40503, 40499 **(3)**