```
> with(numtheory);
Warning, new definition for order
[B, F, GIgcd, J, L, M, bernoulli, bigomega, cfrac, cfracpol, cyclotomic, divisors, euler,
    factorEQ, factorset, fermat, ifactor, ifactors, imagunit, index, integral_basis, invcfrac, invphi,
    isolve, isprime, issqrfree, ithprime, jacobi, kronecker, λ, legendre, mcombine, mersenne,
    minkowski, mipolys, mlog, mobius, mroot, msqrt, nearestp, nextprime, nthconver, nthdenom,
    nthnumer, nthpow, order, pdexpand, ϕ, pprimroot, prevprime, primroot, quadres, rootsunity,
    safeprime, σ, sq2factor, sum2sqr, τ, thue]
> interface(verboseproc=2);
> print(ifactor);
proc(n)
local sol, r, t1;
global `ifactor/bottom`;
option
remember, system, `Copyright (c) 1991 by the University of Waterloo. All rights reserved.`;
    if nargs < 1 or 1 < nargs and not type(args[2], name) then
        ERROR(`invalid arguments`)
    fi;
    if type(n, integer) then
        if 0 < n then sol := 1; r := n elif n < 0 then sol := -1; r := –n else RETURN(0) fi
    elif type(n, fraction) then RETURN(ifactor(op(1, n)) / ifactor(op(2, n)))
    elif type(n, {list, `*`, set, relation}) then RETURN(map(ifactor, n))
    elif type(n, `^`) and type(op(2, n), integer) then RETURN(ifactor(op(1, n))^op(2, n))
    elif type(n, ``(integer)) then RETURN(ifactor(op(1, n)))
    else ERROR(`invalid arguments`)
    fi;
    if assigned(`ifactor/from_signature`[r]) then RETURN(`ifactor/from_signature`[r]) fi;
    t1 := igcd(r, 720720);
    while t1 ≠ 1 do sol := sol*`ifactor/ifact235`(t1); r := iquo(r, t1); t1 := igcd(t1, r) od;
    t1 := igcd(r, 1165429511437014421182564255394118062787055181073690352484362\
        7244292886551970895780845374261270209629822427348440139234569670132\
        5406686013877435618390976369644306706905986206519203898847841821908\
        2943879223019472668430243781122990303492985397077416788992156275405\
        4980997653579451924797222138572272120006081285601711976593494241961\
        1716722790220172291221277043483457977929715054446536085051102596639\
        2537774945829901979588813143194547543583825739867652770670299983480\
        7973358746434146103434351501879897281549965776192088074893821475981\
        1175844174797182002767061333889689987815611105878789730794721580125\
        9320492843992867678444319238233981852191131219418884757870660310007\
        540404320811366733424902110215685045241323250289709233);
    while t1 ≠ 1 do sol := sol*`ifactor/ifact1st`(t1); r := iquo(r, t1); t1 := igcd(t1, r) od;
    if r ≠ 1 then
        if nargs = 1 then
            `ifactor/bottom` := readlib('`ifactor/morrbril`'); t1 := `ifactor/ifact0th`(r)
        else
            `ifactor/bottom` := readlib(`ifactor/`.(args[2]));
            t1 := `ifactor/ifact0th`(r, args[3 .. nargs])
        fi;
        if t1 ≠ FAIL then sol*t1 else FAIL fi
    else sol
    fi
end
```