```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2003
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
>   interface(screenwidth=120);
# egalite de 2vecteurs a facteur pres:
>  egal:=proc(P,Q);
>  a:=expand(P[1]*Q[2]-P[2]*Q[1]);
>  b:=expand(P[3]*Q[2]-P[2]*Q[3]);
>  c:=expand(P[1]*Q[3]-P[3]*Q[1]);
>  if [a,b,c]=[0,0,0] then true; else false; fi;
> end proc;
egal := proc(P, Q)
local a, b, c;
    a := expand(P[1]*Q[2] - P[2]*Q[1]);
    b := expand(P[3]*Q[2] - P[2]*Q[3]);
    c := expand(P[1]*Q[3] - P[3]*Q[1]);
    if [a, b, c] = [0, 0, 0] then true else false end if
end proc

#On cree une procedure simplifier que l'on pourra modifier ensuite:
> simplifier:=proc(R)
> d:=igcd(R[1],R[2],R[3]);
> R/d;end proc;
                        simplifier := proc(R) local d; d := igcd(R[1], R[2], R[3]); R/d end proc

>  #C y^2=x^3+ax^2+bx+c
> pointresiduel:=proc(a,b,c,P,Q)
> C:=-y^2*z+x^3+a*x^2*z+b*x*z^2+c*z^3;
> if egal(P,Q) then
#la tangente est P+tQQ
> QQ:=[subs(x=P[1],y=P[2],z=P[3],diff(C,y)),subs(x=P[1],y=P[2],z=P[3],-diff(C,x)),0];
> M:=expand(P+t*QQ);d:=subs(x=M[1],y=M[2],z=M[3],C);
> sol:=simplify(d/t^2);
> if egal(QQ,[0,1,0]) then R:=QQ; else R:=expand(-coeff(sol,t,1)*P+coeff(sol,t,0)*QQ);fi;
> else
#On parametre (PQ) par P+tQ, 0 et l'infini sont solutions.
> M:=expand(P+t*Q);d:=subs(x=M[1],y=M[2],z=M[3],C);
#sol doit etre un poly de degre 1 puisque 0 et l'infini sont solutions.
> sol:=simplify(d/t);
#ce vecteur doit etre non nul.
> R:=expand(-coeff(sol,t,1)*P+coeff(sol,t,0)*Q);
> fi;
> simplifier(R);
> end proc;
pointresiduel := proc(a, b, c, P, Q)
local C, QQ, M, d, sol, R;
    C := -y^2*z + x^3 + a*x^2*z + b*x*z^2 + c*z^3;
    if egal(P, Q) then
        QQ := [subs(x = P[1], y = P[2], z = P[3], diff(C, y)), subs(x = P[1], y = P[2], z = P[3], -diff(C, x)), 0];
        M := expand(P + t*QQ);
        d := subs(x = M[1], y = M[2], z = M[3], C);
        sol := simplify(d/t^2);
        if egal(QQ, [0, 1, 0]) then R := QQ else R := expand(-coeff(sol, t, 1)*P + coeff(sol, t, 0)*QQ) end if
    else
        M := expand(P + t*Q);
        d := subs(x = M[1], y = M[2], z = M[3], C);
        sol := simplify(d/t);
        R := expand(-coeff(sol, t, 1)*P + coeff(sol, t, 0)*Q)
    end if;
    simplifier(R)
end proc

#exemple:
> P:=[0,0,1];Q:=[1,1,1];omega:=[0,1,0];
                                                P := [0, 0, 1]

                                                Q := [1, 1, 1]

                                            omega := [0, 1, 0]

#verification egal:
>  if egal(P,Q) then print(eg) else  print(noneg) fi;
                                                    noneg

>  if egal(P,P) then print(eg) else  print(noneg) fi;
                                                      eg

>  pointresiduel(1,-1,0,P,Q),pointresiduel(1,-1,0,P,omega),pointresiduel(1,-1,0,P,omega);
                                          [-1, -1, 1], [0, 0, 1], [0, 0, 1]

#addition:
> plus:=proc(a,b,c,P,Q)
> R:=pointresiduel(a,b,c,P,Q);
> R[2]:=-R[2];
> R;
> end proc;
           plus := proc(a, b, c, P, Q) local R; R := pointresiduel(a, b, c, P, Q); R[2] := -R[2]; R end proc

> plus(1,-1,0,omega,omega),plus(1,-1,0,P,omega),plus(1,-1,0,P,P);R:=plus(1,-1,0,P,Q);
Error, (in simplifier) numeric exception: division by zero
                                                 R := [-1, 1, 1]

#On programme la mult par n en O(log n) additions.
> nplus:=proc(a,b,c,P,n)
> Y:=[0,1,0];X:=P;m:=n;
> while m>1 do
>   if type(m,even) then  X:=plus(a,b,c,X,X);m:=m/2;
>   else Y:=plus(a,b,c,X,Y);X:=plus(a,b,c,X,X);m:=(m-1)/2;
>   fi;end do;
>   plus(a,b,c,X,Y);end proc;
nplus := proc(a, b, c, P, n)
local Y, X, m;
    Y := [0, 1, 0];
    X := P;
    m := n;
    while 1 < m do
        if type(m, even) then X := plus(a, b, c, X, X); m := 1/2*m
        else Y := plus(a, b, c, X, Y); X := plus(a, b, c, X, X); m := 1/2*m - 1/2
        end if
    end do;
    plus(a, b, c, X, Y)
end proc

#P est d'ordre 2, Q d'ordre 3, P+Q d'ordre 6
>  nplus(1,-1,0,P,2),nplus(1,-1,0,Q,2),nplus(1,-1,0,Q,3),nplus(1,-1,0,R,3),nplus(1,-1,0,R,2),nplus(1,-1,0,R,6);
Error, (in simplifier) numeric exception: division by zero
>
#Pb de factorisation.
> egalomega:=proc(P,N)
>  Q:=[0,1,0];
>  a:=expand(P[1]*Q[2]-P[2]*Q[1]);
>  b:=expand(P[3]*Q[2]-P[2]*Q[3]);
>  c:=expand(P[1]*Q[3]-P[3]*Q[1]);
>  g:=igcd(a,b,c,N);
>  if g<>1 and g<>N  then print("diviseur",g):true; else false; fi;
> end proc;
egalomega := proc(P, N)
local Q, a, b, c, g;
    Q := [0, 1, 0];
    a := expand(P[1]*Q[2] - P[2]*Q[1]);
    b := expand(P[3]*Q[2] - P[2]*Q[3]);
    c := expand(P[1]*Q[3] - P[3]*Q[1]);
    g := igcd(a, b, c, N);
    if g <> 1 and g <> N then print("diviseur", g); true else false end if
end proc

#On redefinit simplifier pour travailler modulo N
> simplifier:=proc(R)
> d:=igcd(R[1],R[2],R[3],N);
>  if d<>1 and d<>N then print("diviseur",d); break;
> else  R mod N;fi;end proc;
simplifier := proc(R)
local d;
    d := igcd(R[1], R[2], R[3], N); if d <> 1 and d <> N then print("diviseur", d); break else R mod N end if
end proc

>
#On redefinit plus pour qu'il s'arrete si l'on obtient
#omega modulo un diviseur strict de N.
> plus:=proc(a,b,c,P,Q)
> R:=pointresiduel(a,b,c,P,Q);
> R[2]:=-R[2];
> if egalomega(R,N) then break; fi;
> R;
> end proc;
plus := proc(a, b, c, P, Q)
local R;
    R := pointresiduel(a, b, c, P, Q); R[2] := -R[2]; if egalomega(R, N) then break end if; R
end proc

#factorisation d'un entier via les courbes elliptiques
> n:=ilcm(seq(i,i=1..19));
                                                  n := 232792560

#On cherche c tel que (2,1) soit sur y^2=x^3+bx+c
> P:=[2,1,1];N:=nextprime(20000)*nextprime(40000);
                                                  P := [2, 1, 1]

                                                  N := 800620099

> for a from 1 to 3 do for b from 1 to 20
> do nplus(a,b,P[2]^2-P[1]^3-a*P[1]^2-b*P[1],P,n) od:od:
                                                 "diviseur", 20011

Error, (in nplus) break or next outside do loop
> print("trouve en a b"),a,b;
                                                "trouve en a b"

                                                      1, 2

> N:=nextprime(1234567)*nextprime(7654321);n:=ilcm(seq(i,i=1..50));
                                                  N := 9449868410449

                                              n := 30990445042459967064 00

> for a from 1 to 3 do for b from 1 to 20
> do nplus(a,b,P[2]^2-P[1]^3-a*P[1]^2-b*P[1],P,n) od:od:
bytes used=4000324, alloc=3276200, time=0.11
bytes used=8000776, alloc=4062488, time=0.23
                                                "diviseur", 1234577

Error, (in nplus) break or next outside do loop
> print("trouve en a b"),a,b;
                                                "trouve en a b"

                                                      1, 2

> quit
bytes used=8048976, alloc=4062488, time=0.23
```