

```

#Additional 2
>
> f:=x^5+3*x^4*y+3*x^3*y^2-2*x^4*y^2+x^2*y^3-6*x^3*y^3-6*x^2*y^4+x^3*y^4-2*x*y^5+3*x^2*y^5+3*x*y^6+y^7:
> fx:=diff(f,x):
> fy:=diff(f,y):
>
#To get g1=gcd(f,fx) we apply Thm 11
> G:=Groebner[Basis]([t*f,(1+t)*fx],plex(t,x,y)):
> map(indets,G);
      [{x, y}, {x, y, t}, {x, y, t}]

#by Prop 9 this is the LCM of f and fx
> LCM:=G[1];
      3 6      2 7      8      9      4 4      3 5      2 6
LCM := -3 x y - 9 x y - 9 x y - 3 y + 11 x y + 35 x y + 39 x y
      7      8      5 2      4 3      3 4      2 5      6
      + 17 x y + 2 y - 13 x y - 43 x y - 51 x y - 25 x y - 4 x y
      6      5      4 2      3 3      2 4
      + 5 x + 17 x y + 21 x y + 11 x y + 2 x y

#so the gcd of f and fx is:
> g1:=simplify(f*fx/LCM);
      3      2 2      2      3      2      4
g1 := x - x y + 2 x y - 2 x y + y x - y

>
#To get g = gcd(g1,fy) = gcd(f,fx,fy) we apply Thm 11
> G:=Groebner[Basis]([t*g1,(1+t)*fy],plex(t,x,y)):
> map(indets,G);
      [{x, y}, {x, y, t}]

#by Prop 9 this is the LCM of gcd(f,fx) and fy
> LCM:=G[1];
      4      3      4      2 2      3 2      2 3      3 3
LCM := -3 x - 6 x y + 4 x y - 3 x y + 18 x y + 24 x y - 4 x y
      4      2 4      5      6
      + 10 x y - 15 x y - 18 x y - 7 y

#so the gcd of gcd(f,fx) and fy is:
> g:=simplify(g1*fy/LCM);
      2 2      3      4      3      2      2
g := x y + 2 x y + y - x - 2 x y - y x

>
#let us check the result
> eval(GCD);
      proc(x::seq(algebraic)) foldr(gcd, 0, x) end proc

> GCD(f,fx,fy);
      3      2 2      2      3      2      4
x - x y + 2 x y - 2 x y + y x - y

#(this will only be equal upto a scalar

```