def triangulation(S):

T=[]

n=0

while S!=[]:

m=max([s.lm() for s in S])

L = [s for s in S if s.lm() == m]

S = [s for s in S if s.lm() < m]

g1=L[0]

T.append(g1)

for g in L[1:]:

g=g1.lc()\*g-g.lc()\*g1

if g.degree()==0:

T.append(g)

break

elif g.degree()==1:

S.append(g)

return T

def tsolve(T):

T.reverse()

D={}

while T!=[]:

g=T[0]

D[g.lm()] = -(g-g.lt())/g.lc()

T=[t.subs(D) for t in T[1:]]

return D