def ostrogradski(f,x):

g=SR(f).numerator()

h=SR(f).denominator()

# Раскладываем знаменатель на множители

L=list(ZZ[x](h).factor())

# Составляем знаменатель алгебраической части

hh=prod([p^(m-1) for (p,m) in L])

n=hh.degree()

if n==0:

return [0,f]

else:

# Составляем числитель алгебраической части

A=var(['A'+str(i) for i in range(n)])

gg=sum([A[i]\*x^i for i in range(n)])

# Составляем знаменатель log части

h3=ZZ[x](prod([p for (p,m) in L]))

# Составляем числитель log части

m=h3.degree()

B=var(['B'+str(i) for i in range(m)])

g3=sum([B[i]\*x^i for i in range(m)])

# Составляем выражение, которое должно быть равно нулю

F=ZZ[A+B][x]((diff(SR(gg/hh),x)+SR(g3/h3)-f).numerator())

# Работаем со СЛАУ

S=tsolve(triangulation([QQ[A+B](eq) for eq in F.coefficients()]))

# Список: алгебраическая часть, log часть

return [(gg).subs(S)/hh, (g3).subs(S)/h3]

def radical(a,r):

if r==True:

return AA(a).radical\_expression()

else:

return AA(a)

def pfdintegral(f,x,r):

g=SR(f).numerator()

h=SR(f).denominator()

if AA[x](h).degree()==1:

return radical(g,r)\*ln(abs(x+radical(h.subs(x=0),r)))

else:

b0=radical(g.subs(x=0),r)

b1=radical(diff(g,x),r)

a0=radical(h.subs(x=0),r)

a1=radical(diff(h,x).subs(x=0),r)

s=radical(sqrt(-a1^2 + 4\*a0),r)

return 1/2\*b1\*log(a1\*x + x^2 + a0) - (a1\*b1 - 2\*b0)\*arctan((a1 + 2\*x)/s)

def rac\_integral(f,x, list=False, radical=True):

# Шаг 1. Приведение к правильной дроби

g=SR(f).numerator()

h=SR(f).denominator()

[U,r]=QQ[x](g).quo\_rem(QQ[x](h))

# Шаг 2. Отыскание алгебраической части

[A,L]=ostrogradski(SR(r/h),x)

# Шаг 3. Интегрирование log части

pfd=FractionField(AA[x])(L).partial\_fraction\_decomposition()[1]

# Сборка ответа: интеграл многочлена + алгебраическая часть + интеграл от логans= [integral(U,x), A]+[pfdintegral(i,x,radical) for i in pfd]

if list==True:

return ans

else:

return sum(ans)