

```

from math import *
def discriminant(L):
    return L[1]**2-4*L[0]*L[2]

def racines(L):
    d=discriminant(L)
    if d>0:
        return [(-L[1]-sqrt(d))/(2*L[0]),(-L[1]+sqrt(d))/(2*L[0])]
    elif d==0:
        return [-L[1]/(2*L[0])]
    else:
        return "aucune racine réelle"

def sgnpoly(L):
    d=discriminant(L)
    if d>0:
        if L[0]>0:
            return ("positif sauf entre ses racines",[(-L[1]-sqrt(d))/(2*L[0]),(-L[1]+sqrt(d))/(2*L[0])])
        if L[0]<0:
            return ("négatif sauf entre ses racines",[(-L[1]-sqrt(d))/(2*L[0]),(-L[1]+sqrt(d))/(2*L[0])])
    elif d==0:
        if L[0]>0:
            return ("strictement positif et s'annule en ",[-L[1]/(2*L[0])])
        if L[0]<0:
            return ("strictement négatif et s'annule en ",[-L[1]/(2*L[0])])
    else:
        if L[0]>0:
            return ("strictement positif pour tout x réel")
        if L[0]<0:
            return ("strictement négatif pour tout x réel")

```