Function: citire\_coef(coef, n):

for i in range(n):

Scrie "coef[",i,"] <- ", sep <- '', end <- '')

coef.append(int(input()))

Scrie )

return coef

Function: cautare\_div\_termen\_liber(num):

lista\_div <- []

for i in range(1,num + 1):

if num % i = 0:

lista\_div.append(i)

lista\_div.append((-i))

return lista\_div

Function: cautare\_div\_termen\_grad\_maxim(num):

lista\_div <- []

for i in range(1,abs(num) + 1):

if num % i = 0:

lista\_div.append(i)

lista\_div.append(-i)

return lista\_div

Function: radacini(div\_tl, div\_tm)

lista\_radacini <- []

for i in div\_tl:

for j in div\_tm:

lista\_radacini.append((i,j))

return lista\_radacini

Function: cmmdc(a,b)

while (b !<- 0):

r <- a % b

a <- b

b <- r

return a

Function: simplificare(numarator1, numitor1)

a <- numarator1

b <- numitor1

div <- cmmdc(numarator1,numitor1)

return numarator1//div, numitor1//div

Function: citire\_tablou(tablou\_coef, c)

for i in range(c):

Scrie 'tabl\_coef[', i, '] <- ', end <- '')

tablou\_coef.append(int(input()))

Function: adunare(numarator1, numitor1, numarator2, numitor2):

if numarator1 = 0:

return numarator2,numitor2

else:

numarator1 <- (numitor2 \* numarator1) + (numarator2 \* numitor1)

numitor1 <- numitor1 \* numitor2

if numarator1 = 0:

return 0, 0

return numarator1, numitor1

Function: scadere(numarator1, numitor1, numarator2, numitor2)

if numarator1 = 0:

return -numarator2, numitor2

else:

numarator1 <- (numitor2 \* numarator1) - abs((numarator2 \* numitor1))

numitor1 <- numitor1 \* numitor2

if numarator1 = 0:

return 0, 0

return numarator1, numitor1

Function: calculare\_polinom(lista\_radacini, coef)

lista\_rationali <- []

pentru k <- 0 , k < lungime(lista\_radacini)):

sum\_numarator <- 0

sum\_numitor <- 0

pentru i <- n-1, 0, -1)

numa <- lista\_radacini[k][0]

numi <- lista\_radacini[k][1]

numa <- numa \*\* i

numi <- numi \*\* i

numa <- numa \* coef[i]

numa, numi <- simplificare(numa, numi)

if numa < 0:

sum\_numarator, sum\_numitor <- scadere(sum\_numarator, sum\_numitor, numa, numi)

else:

sum\_numarator, sum\_numitor <- adunare(sum\_numarator, sum\_numitor, numa, numi)

if coef[0] < 0:

sum\_numarator, sum\_numitor <- scadere(sum\_numarator, sum\_numitor, -1 \* coef[0], 1)

else:

sum\_numarator, sum\_numitor <- adunare(sum\_numarator, sum\_numitor, coef[0], 1)

if sum\_numarator = 0:

lista\_rationali.append([lista\_radacini[k][0], lista\_radacini[k][1]])

return lista\_rationali

Scrie "Cati coeficienti sa aiba polinomul?(incluzand termenul liber): ", end <- '')

n <- 3

Scrie "Scrieti coeficientii, termenul de pe ultima pozitie va fi termenul liber:"

coef <- [8,-10,3]

# coef <- citire\_coef(coef, n)

div\_tl <- cautare\_div\_termen\_liber(coef[len(coef) - 1])

div\_tm <- cautare\_div\_termen\_grad\_maxim(coef[0])

lista\_radacini <- radacini(div\_tl, div\_tm)

Scrie lista\_radacini

Scrie "Lista cu divizorii termenului de grad maxim (", coef[0], ") este: ", div\_tm)

Scrie "Lista cu divizorii termenului liber (", coef[len(coef)-1], ") este:", div\_tl)

coef <- coef[::-1]

rez <- calculare\_polinom(lista\_radacini, coef)

# Am lucrat cu modul iar pentru un rezultat corect de forma (a,b) imi afiseaza si (-a,-b), inca nu stiu cum

# sa rezolv aceasta problema de afisare, asa ca le-am convertit pe toate la valoarea absoluta

for i in range(len(rez)):

if rez[i][0] < 0 and rez[i][1] < 0

rez[i][0] <- abs(rez[i][0])

rez[i][1] <- abs(rez[i][1])

Scrie rez