def f(x):

    return x\*\*4+x\*\*3+5\*x\*\*2+4\*x+4

p0=float(input('Enter value of initial approximate root po'))

a0=float(input('Enter value of a0'))

a1=float(input('Enter value of a1'))

a2=float(input('Enter value of a2'))

a3=float(input('Enter value of a3'))

a4=float(input('Enter value of a4'))

i=int(input('Enter no of iteration i'))

count=1

while (count<=i):

  b0=a0

  b1=a1+p0\*b0

  b2=a2+p0\*b1

  b3=a3+p0\*b2

  b4=a4+p0\*b3

  c0=b0

  c1=b1+p0\*c0

  c2=b2+p0\*c1

  c3=b3+p0\*c2

  p1=p0-(b4/c3)

  print("iteration :",count)

  print("new root is",p1)

  p0=p1

  count=count+1

# Program for Birge-Vieta Method

n=int(input('Enter order of equation:'))

a1=input("Enter coefficients:")

p=input('Enter initial guess:')

it=int(input('Enter number of iterations required:'))

a=list(map(float,a1.split()))

p=list(map(float,p.split()))

print(p)

for k in range(0,it):

b=[]; c=[];

b.append(a[0]);

c.append(b[0]);

j=1;

for j in range(1,n+1):

b.append(a[j]+p[k]\*b[j-1])

c.append(b[j]+p[k]\*c[j-1])

#print(b,c)

p.append(p[k]-b[n]/c[n-1])

print("Root of equation at iteration",k+1,"is", p[k+1])