**import** random

**def** LCM(l): *# 최소공배수*

myl = l.copy()

**for** i **in** range(len(l) - 1):

lcm = myl[i] \* myl[i + 1] / GCD(myl[i:i + 2])

myl[i + 1] = lcm

**return** int(lcm)

**def** GCD(l): *# 최대공약수*

**for** k **in** range(min(l), 0, -1):

b = 0

**for** j **in** range(len(l)):

**if** l[j] % k == 0:

b += 1

**if** b == len(l):

**return** k

**def** main():

a = []

**for** i **in** range(10):

a.append(random.randint(1, 100)) *# 1에서 100 중 정수 하나*

print(a)

lcm\_value = LCM(a)

gcd\_value = GCD(a)

print(**'최소공배수: '**, lcm\_value, **' 최대공약수: '**, gcd\_value)

**if** \_\_name\_\_ == **'\_\_main\_\_'**:

main()

--------------------------

**def** check\_prime(num):

**if** num == 1 :

**return** 0

**for** i **in** range(2, num) :

**if** num%i == 0 :

**return** 0

**return** 1

**def** main():

a = 13

b = 15

**if** check\_prime(a):

print(str(a)+**'는 소수입니다.'**)

**else**:

print(str(a)+**'는 소수가 아닙니다.'**)

**if** check\_prime(b):

print(str(b)+**'는 소수입니다.'**)

**else**:

print(str(b)+**'는 소수가 아닙니다.'**)

**if** \_\_name\_\_==**'\_\_main\_\_'**:

main()

**def** add\_comma(val):

s=str(val)

**if** len(s)>3 :

**for** i **in** range(-len(s),0):

**if** i%3==0 :

s=s[:i]+**','**+s[i:]

**return** s

**def** main():

comma\_added\_1234 = add\_comma(1234)

comma\_added\_12345678 = add\_comma(12345678)

comma\_added\_12 = add\_comma(12)

print(comma\_added\_1234) *# ‘1,234’*

print(comma\_added\_12345678) *# ‘12,345,678’*

print(comma\_added\_12) *# ‘12’*

**if** \_\_name\_\_==**'\_\_main\_\_'**:

main()

**def** mean\_and\_var(\*val):

hap1=0;hap2=0;res1=0;res2=0

n=len(val)

**for** i **in** val :

hap1+=i[0]

hap2+=i[1]

m1 = hap1 / n; m2 = hap2 / n

m=(m1, m2)

**for** i **in** val :

res1+=((i[0]-m1)\*\*2)

res2+=((i[1]-m2)\*\*2)

va1 = res1 / n; va2 = res2 / n

var=(va1, va2)

**return** m, var

**def** main():

v1=(0, 1)

v2=(0.5, 0.5)

v3=(1, 0)

m, var = mean\_and\_var(v1, v2, v3)

print(**'평균: '**, m, **'분산: '**, var)

**if** \_\_name\_\_==**'\_\_main\_\_'**:

main()