문제3 아래와 같이 2차원 행렬 두개의 내적을 구하는 함수 lt\_dot 함수를 만드시오

추가조건

-행렬의 요소 검사 후 예외처리

-행렬의 크기가 잘못된 경우 예외 처리

실행결과

lt1=[[1,2],[3,4]]

lt2=[[5],[6]]

print(lt\_dot(lt1,lt2)) #[[17],[39]]

답

def ltall(v):

if type(v)==int or type(v)==float:

return True

elif type(v)==list or type(v)==tuple:

lt=[ltall(i) for i in v]

return True if all(lt) else False

else :

return False

def lt\_dot(lt1,lt2):

if not ltall(lt1) or not ltall(lt2):

print("error")

if len(lt1[0])!=len(lt2) :

print("error")

return False

lt3=[]

for i in (lt1):

tmp=[]

for j in (lt\_trans(lt2)):

tmp.append(sum(lt\_mul(i,j)))

lt3.append(tmp)

return lt3

def lt\_trans(lt):

row=len(lt)

col=len(lt[0])

lt2=[]

for i in range(col):

tmp=[]

for j in range(row):

tmp.append(lt[j][i])

lt2.append(tmp)

return lt2

def lt\_mul(ltA,ltB):

ltC=[ltA[i]\*ltB[i] for i in range(len(ltA))]

return ltC

lt1=[[1,2],[3,4]]

lt2=[[5],[6]]

print(lt\_dot(lt1,lt2))

문제 5

[1,2,3,4,5] 리스트에 3을 곱한 [3,6,9,12,15] 리스트 결과를 만들고, 리스트 성분이 짝수인 경우를 제외한 [3,9,15] 리스트를 만드는 방법 두 가지를 각각 작성해보자

방법1

lt=[1,2,3,4,5]

lt2=[kk\*3 for kk in lt]

lt3=[k for k in lt2 if k%2==1]

print(lt3)

방법2

def de1(a):

if a%2==0:

return False

else:

return True

def de2(a):

return a\*3

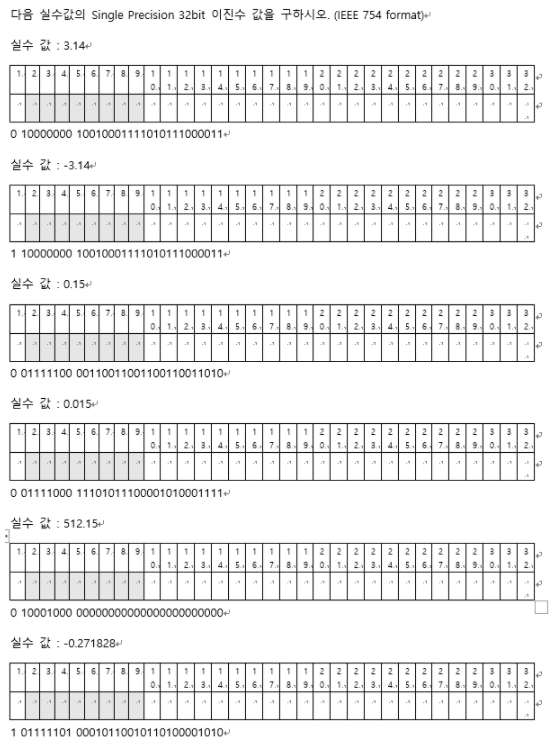
lt=[1,2,3,4,5]

lt2=list(map(de2,lt))

lt3=list(filter(de1,lt2))

print(lt3)

HW8문제



<코드>

num=float(input("실수값 :"))

if num<0:

a=1

else:

a=0

num=abs(num) #음수이거나 양수이거나 a만 달라지므로 절댓값

u=int(num) #10진법일 때 소수점 위

d=num-u #10진법일 때 소수점 아래

ondot=str(bin(u)).replace('0b','') #2진법일 때 소수점 위

underdot='' #2진법일 때 소수점 아래

for i in range(30): #소수점위가 0일 때 소수점아래 숫자가 23개 이상 필요하므로

if d\*2>=1: #여유있게 30개

underdot+='1'

d=d\*2-1

else :

underdot+='0'

d\*=2

if u!=0:

alpha=len(ondot)-1

c=ondot[1:]+underdot[:23-alpha]

else:

alpha=0

for i in underdot:

alpha-=1

if i=='1':

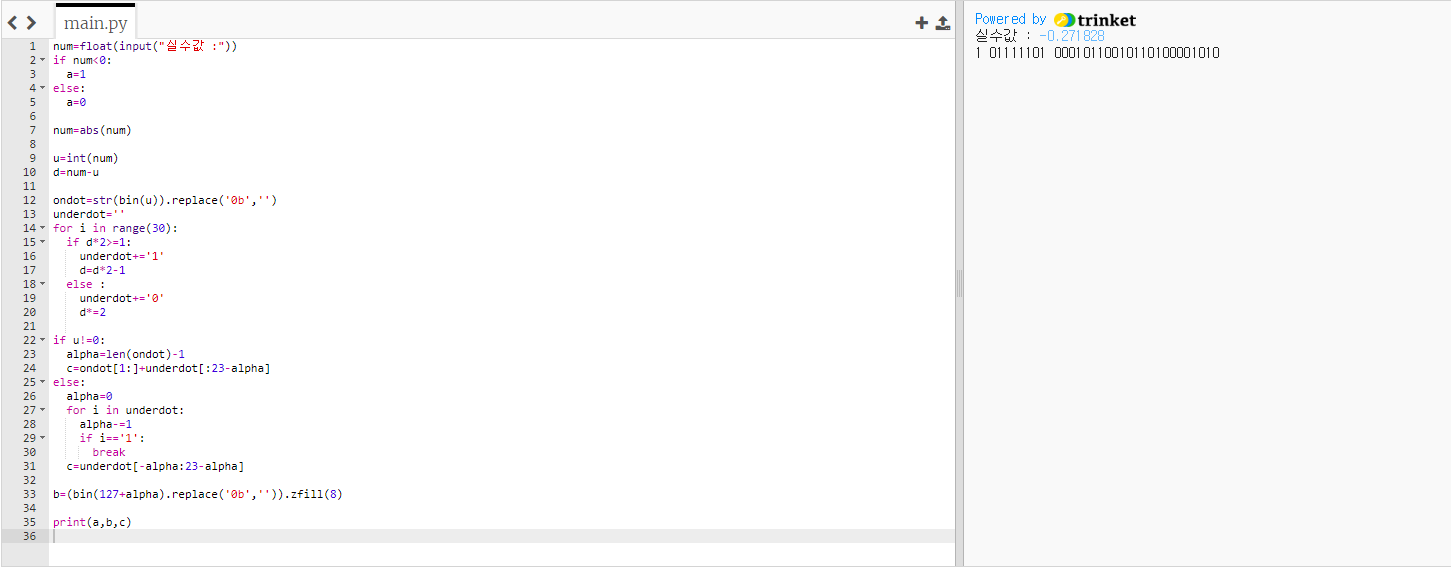
break

c=underdot[-alpha:23-alpha]

b=(bin(127+alpha).replace('0b','')).zfill(8)

print(a,b,c)

<실행화면>



<구현내용설명>

출력되는 숫자들을 a, b, c로 나누어 연산했습니다.

alpha: 입력 받은 숫자를 2진법으로 변환 후 그 수를 1.~~~~\*으로 나타낼 때 n

a: 입력 받은 숫자가 음수면 1 아니면 0

b: 127+alpha를 2진수로 나타냄

c: 입력 받은 숫자를 2진법으로 변환 후 그 수를 1.~~~~\*으로 나타낼 때 ~~~~~(23자리)

입력 받은 소수점 위의 숫자가 0이 아니면, 소수점 위의 숫자에서 첫번째 숫자를 제외한 숫자들 + 소수점 아래숫자 (23-alpha)개

입력 받은 소수점 위의 숫자가 0이면, 소수점 아래의 숫자 중 첫번째 1을 제외한 그 아래 23개 숫자