DAY 9

2105A41202

Program:

def demo(s):

l=[]

for i in s:

if i.isalpha():

l.append(i)

else:

spc=i

idxspc=s.index(i)

l.reverse()

l.insert(idxspc,spc)

return''.join(l)

s=input()

print(demo(s))

Program:

#input=7564168

# example:separete odd place integer:5 4 6

#you have to return a 4 digit OTP by squaring the digits.

#digits from above ex:5 4 6,squares:25,16,36 so OTP to be returned is first four digits:2516

n=input()#7564168

op=''

for i in range(1,len(n),2):

op+=str(int(n[i])\*\*2)

print(op[:4])

Binary Search:

def binary\_search\_ceil(arr,target):

left,right=0,len(arr)

#ceil=float('int')# +ve infinity value,it will behave as if it is infinity

while left <=right:

mid=left+(right-left) //2

if arr[mid]==target:

return arr[mid]

elif arr[mid]<target:

left=mid+1

else:

ceil=arr[mid]

right=mid-1

return ceil

print(binary\_search\_ceil([1,2,8,10,12,19],7))

Program:

n,k=map(int,input().split())

lst=[]

for i in range(n):

lst.append(int(input()))

s,c=set(lst),0

for i in s:

if lst.count(i)>=k:

c+=1

print(c)

Program:

def binary\_search\_floor(arr,target):

left,right=0,len(arr)-1

floor=-1

while left <=right:

mid=left+(right-left) //2

if arr[mid]==target:

return arr[mid]

elif arr[mid]<target:

floor=arr[mid]

left=mid+1

else:

right=mid-1

return floor

print(binary\_search\_floor([1,2,8,10,12,19],7))

Program:

def demo(s): #2,5,1,4,3,2,7,8

los=s.split(',')

idxpof=los.index('4')

idxpos=los.index('7')

n1,n2=0,''

for i in los[:idxpof]+los[idxpos+1:]:

n1+=int(i)

for i in los[idxpof:idxpos+1]:

n2+=i

return(n1+int(n2))

s=input()

if \_name=='main\_':

print(demo(s))

Program:

def demo(s): #()))

c=0

l=[]

for i in s:

if i=='[' or i=='{' or i=='(':

l.append(i)

c+=1

continue

if len(l)==0:

return c+=1

temp=l.pop()

if temp=='(' and i==')':

c+=1

elif temp=='[' and i==']':

c+=1

elif temp=='{' and i=='}':

c+=1

else:

return c+1