def count\_frequencies(arr):

freq={}

for num in arr:

freq[num]=freq.get(num,0)+1

return freq

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

frequencies=count\_frequencies(arr)

for i,count in frequencies.items():

print(i,count)

class Solution:

# Function to count the frequency of all elements from 1 to N in the array.

def frequencyCount(self, arr):

freq={}

ans=[]

n=len(arr)

for i in arr:

freq[i]=freq.get(i,0)+1

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

ans.append(freq.get(i,0))

return ans

n=int(input())

for i in range(n):

for j in range(n-i-1):

print(" ",end=" ")

for j in range(i+1):

print("\*",end=" ")

print()

n=int(input())

for i in range(n):

for j in range(n-i-1):

print(" ",end=" ")

for j in range(i+1):

print("\*",end=" ")

print()

n=int(input())

for i in range(n):

for j in range(i):

print(" ",end=" ")

for j in range(n-i):

print("\*",end=" ")

print()

n=6

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

for y in range(0,x):

print(str((x+y)%2)+"",end="")

print()

n=5

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

print(" "\*(n-i),end="")

x=1

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

print(x,end="")

x=x\*(i-j)//j

print()

for i in range(1,5):

for j in range(4-i):

print(end="")

for j in range(1,i\*2):

print("\*",end=" ")

print()

for i in range(3,0,-1):

for j in range(4-i):

print(end=" ")

for j in range(1,i\*2):

print("\*",end="")

print()

def floyd\_triangle(n):

num=1

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

for k in range(n-i,0,-1):

print(end=" ")

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

print("%2"%num,end=" ")

num=num+1

print("")

floyd\_triangle(6)

class Solution:

#Function to rotate an array by d elements in counter-clockwise direction.

def rotateArr(self, arr, d):

n=len(arr)

temp=[0]\*n

d=d%n

for i in range(len(arr)):

temp[(n-d+i)%n]=arr[i]

for i in range(len(arr)):

arr[i]=temp[i]

return arr

def reverse(arr, start, end):

while start < end:

arr[start], arr[end] = arr[end], arr[start]

start += 1

end -= 1

def rotate\_array(arr, k):

n = len(arr)

k = k % n

reverse(arr, 0, n - 1)

reverse(arr, 0, k - 1)

reverse(arr, k, n - 1)

return arr