**Staircase**

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the staircase function below.*

**def** staircase(n):

**for** i **in** **range**(1,n+1):

**print**((n-1)\*" "+i\*"#")

        n -=1

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

    n = **int**(**input**())

    staircase(n)

# A Very Big Sum

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the aVeryBigSum function below.*

**def** aVeryBigSum(ar):

    result = 0

**for** i **in** ar:

        result += i

**return** result

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    ar\_count = **int**(**input**())

    ar = **list**(**map**(**int**, **input**().rstrip().split()))

    result = aVeryBigSum(ar)

    fptr.write(**str**(result) + '\n')

    fptr.close()

# Simple Array Sum

**import** os

**import** sys

**def** simpleArraySum(ar):

    result = 0

**for** i **in** ar:

        result += i

**return** result

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    ar\_count = **int**(**input**())

    ar = **list**(**map**(**int**, **input**().rstrip().split()))

    result = simpleArraySum(ar)

    fptr.write(**str**(result) + '\n')

    fptr.close()

# Compare the Triplets

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the compareTriplets function below.*

**def** compareTriplets(a, b):

    alice = 0

    bob = 0

**for** i **in** **range**(**len**(a)):

**if** a[i] > b[i]:

            alice += 1

**if** b[i] > a[i]:

            bob += 1

**else**:

**pass**

    result = [**str**(alice),**str**(bob)]

**return** result

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    a = **list**(**map**(**int**, **input**().rstrip().split()))

    b = **list**(**map**(**int**, **input**().rstrip().split()))

    result = compareTriplets(a, b)

    fptr.write(' '.join(**map**(**str**, result)))

    fptr.write('\n')

    fptr.close()

# Diagonal Difference

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

**def** diagonalDifference(arr):

    leftDiagonal = 0

    rightDiagonal = 0

    l = 0

**for** i **in** **range**(0,n):

        leftDiagonal += arr[i][i]

        rightDiagonal += arr[i][n-i-1]

**return** **abs**(leftDiagonal-rightDiagonal)

*# Write your code here*

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    n = **int**(**input**().strip())

    arr = []

**for** \_ **in** **range**(n):

        arr.append(**list**(**map**(**int**, **input**().rstrip().split())))

    result = diagonalDifference(arr)

    fptr.write(**str**(result) + '\n')

    fptr.close()

# Plus Minus

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the plusMinus function below.*

**def** plusMinus(arr):

    positiveCount = 0

    negativeCount = 0

    zeroCount = 0

**for** i **in** arr:

**if** **int**(i) == 0:

            zeroCount += 1

**if** **int**(i) > 0:

            positiveCount += 1

**if** **int**(i) < 0:

            negativeCount += 1

**print**(positiveCount/n,'\n',negativeCount/n,'\n',zeroCount/n )

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

    n = **int**(**input**())

    arr = **list**(**map**(**int**, **input**().rstrip().split()))

    plusMinus(arr)

**Mini-Max Sum**

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the miniMaxSum function below.*

**def** miniMaxSum(arr):

**print**(**sum**(arr) - **max**(arr), **sum**(arr) - **min**(arr))

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

    arr = **list**(**map**(**int**, **input**().rstrip().split()))

    miniMaxSum(arr)

**Birthday Cake Candles**

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*#*

*# Complete the 'birthdayCakeCandles' function below.*

*#*

*# The function is expected to return an INTEGER.*

*# The function accepts INTEGER\_ARRAY candles as parameter.*

*#*

**def** birthdayCakeCandles(candles):

    candles.sort()

    x = candles[-1]

**return** (candles.count(x))

*# Write your code here*

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    candles\_count = **int**(**input**().strip())

    candles = **list**(**map**(**int**, **input**().rstrip().split()))

    result = birthdayCakeCandles(candles)

    fptr.write(**str**(result) + '\n')

    fptr.close()

**def** stringAnagram(dictionary, query):

*# Write your code here*

    result = []

**for** i **in** query:

        list1 = **list**(i)

        list1.sort()

        count = 0

**for** j **in** dictionary:

                list2 = **list**(j)

                list2.sort()

**if** list1 == list2:

                    count =count + 1

        result.append(count)

**return** result

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

    fptr = **open**(os.environ['OUTPUT\_PATH'], 'w')

    dictionary\_count = **int**(**input**().strip())

    dictionary = []

**for** \_ **in** **range**(dictionary\_count):

        dictionary\_item = **input**()

        dictionary.append(dictionary\_item)

    query\_count = **int**(**input**().strip())

    query = []

**for** \_ **in** **range**(query\_count):

        query\_item = **input**()

        query.append(query\_item)

    result = stringAnagram(dictionary, query)

    fptr.write('\n'.join(**map**(**str**, result)))

    fptr.write('\n')

    fptr.close()