**Algorithmic Problem Solving 2020**

**Q-Box Assignment Set**

Student Name: Gourav V Jamakhandi

SRN: 01FE17BCS246

Branch: CSE

**Question 01**

Title: Buoy Range

Level: Easy

Concepts Tested: Co-ordinate Geometry, Arrays

**Problem Statement:**

The Government is deploying Tsunami Buoys in the Indian Ocean in order to provide enhanced tsunami warning system. The buoys were anchored at certain co-ordinates in the ocean. It was found that the area covered by the buoys form a regular polygon. Can you determine the coverage area of the buoys given the ordered co-ordinates of the buoys?

**Input Format:**

The first line contains an integer, T, denoting the number of test cases.

The first line of each test case contains an integer, N, denoting the number of buoys.

The second line contains N space-separated integers x1, x2,…,xN denoting the ordered X co-ordinates.

The third line contains N space-separated integers y1, y2,…,yN denoting the ordered Y co-ordinates.

**Constraints:**

The input co-ordinates are ordered which means that the coordinates are given either in clockwise manner or anticlockwise from first vertex to last.

1<=T<=1000

4<=N<=100

-200 < X[i], Y[i] < 600

**Output Format:**

Print T floating points values (one per line) denoting the area in square units that is covered by the buoys.

**Solution:**

# Python3 Solution

import sys

def calculateArea(X, Y, n):

area = 0

j = n - 1

for i in range(n):

area += (X[j] + X[i]) \* (Y[j] - Y[i])

j = i

return abs(area / 2.0)

t = int(input())

while t > 0:

a = int(input())

b = list(map(int,sys.stdin.readline().split()))

c = list(map(int,sys.stdin.readline().split()))

print(calculateArea(b,c,a))

t-=1

**Sample Test Cases:**

**Sample Input 0**

2

4

-5 5 5 -5

2 2 -2 -2

6

180 323 390 347 216 170

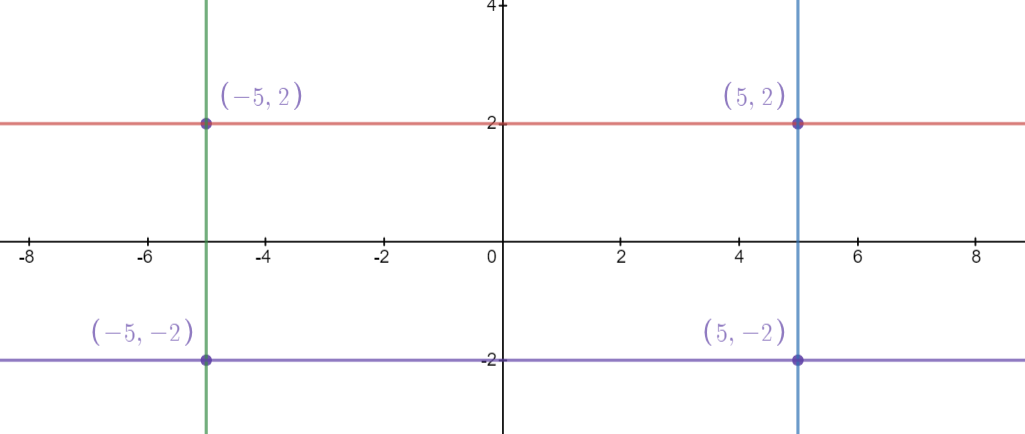
149 122 209 349 333 267

**Sample Output 0**

40.0

37923.0

For the first case, the shape formed by the buoys is a rectangle with l=10 and b=4. Thus, area will be l\*b=4\*10 = 40.0 sq. units.



Similarly, for the second case, the area of the generated hexagon will be 37923.0 sq. units

**Test Cases:**

Test Cases provided in qbox-1.zip (Hackerrank zip format).

No. of Test Cases: **11** (excluding sample)

Uncompressed File Size: **7.37mb**

Compressed File Size: **2.66mb**

**\***Use Challenge Slug **buoy-range** on hackerrank to add the challenge to any contest.

**Question 02**

Title: ADFGVX

Level: Medium

Concepts Tested: String Decryption, 2-D Array Manipulation, Matrix Transposition, Character Array Sorting.

**Problem Statement:**

ADFGVX cipher was a field cipher used by the German Army on the Western Front during World War I to send secret messages. The cipher comprises of a fractionating transposition cipher which combined a modified Polybius square with a single columnar transposition.

The cipher encrypts text and outputs a string containing only the letters A, D, F, G, V and X. The cipher uses a Polybius square along with a transposition key in order to encrypt the string.

The spies have obtained the Polybius square and the key, can you help the Allied Forces decrypt the German messages?

For more info, refer: <https://en.wikipedia.org/wiki/ADFGVX_cipher#ADFGVX>

**Input Format:**

The first line contains an integer, T, denoting the number of test cases.

The first 6 lines of each test case contains a 6x6 Polybius square P, a space separated character matrix made up of letters A-Z and numbers 0-9.

The next line contains a string which represents the transpositional key K.

The next line contains the encrypted message M made up only of the letters A, D, F, G, V and X.

**Constraints:**

The Polybius square P is a square matrix of size 6x6 consisting of characters A-Z and 0-9.

T = 8

2 <= len(K) <= 11

The encrypted message M is made up of letters A, D, F, G, V, X and '-'.

‘-‘ is used as an empty position filler in the generated matrix. Ensure to remove '-' at the end of transposition step during decryption.

**Output Format:**

Print T strings (one per line) denoting the decrypted message.

**Solution:**

# Python3 Solution

def key\_sort(key):

key = list(key)

key.sort()

return "".join(key)

def spread(arr):

res = []

for i in arr:

if isinstance(i,list) or isinstance(i,tuple):

res.extend(i)

else:

res.append(i)

return res

def decipher(matrix,key,cipher):

adfgvx = {"A":0,"D":1,"F":2,"G":3,"V":4,"X":5}

decipher= ""

# Get sorted key

skey = key\_sort(key)

# Generate key table

key\_table = []

for i in range(0,len(cipher),len(cipher)//len(key)):

key\_table.append(list(cipher[i:i+len(cipher)//len(key)]))

# Transpose key table

key\_table = list(zip(\*key\_table))

key\_table.insert(0,list(skey))

# Rearrange key table based on key

key\_table = list(zip(\*key\_table))

kt2 = [0]\*len(key)

for i in key\_table:

kt2[key.index(i[0])]=i

key\_table = list(zip(\*kt2))

# Get encoded string

key\_table.pop(0)

key\_table = "".join(spread(key\_table))

key\_table = key\_table.replace('-','')

for i in range(0,len(key\_table),2):

decipher+=(matrix[adfgvx[key\_table[i]]][adfgvx[key\_table[i+1]]])

return decipher

t = int(input())

for \_ in range(t):

matrix = []

for i in range(6):

matrix.append(input().split())

key = input()

enc = input()

print(decipher(matrix,key,enc))

**Sample Test Cases:**

**Sample Input 0**

1

P M D 1 O W

T C Z J Q R

X 8 2 H S A

0 E 9 L V K

G 7 Y 4 B U

6 F 3 N I 5

GERMANY

DXF-XDAXFXDXAGF-AFX-DDAGFXA-

**Sample Output 0**

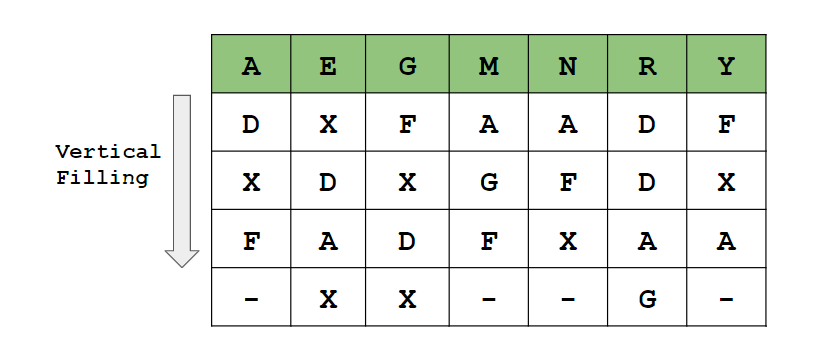
ATTACKATDAWN

**Explanation 0**

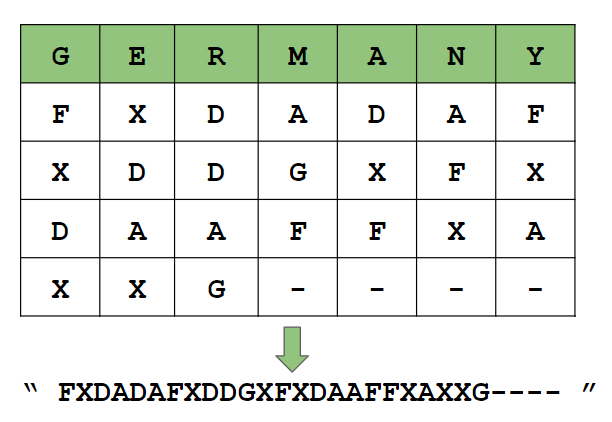
Step 1: Begin by sorting the key alphabetically.



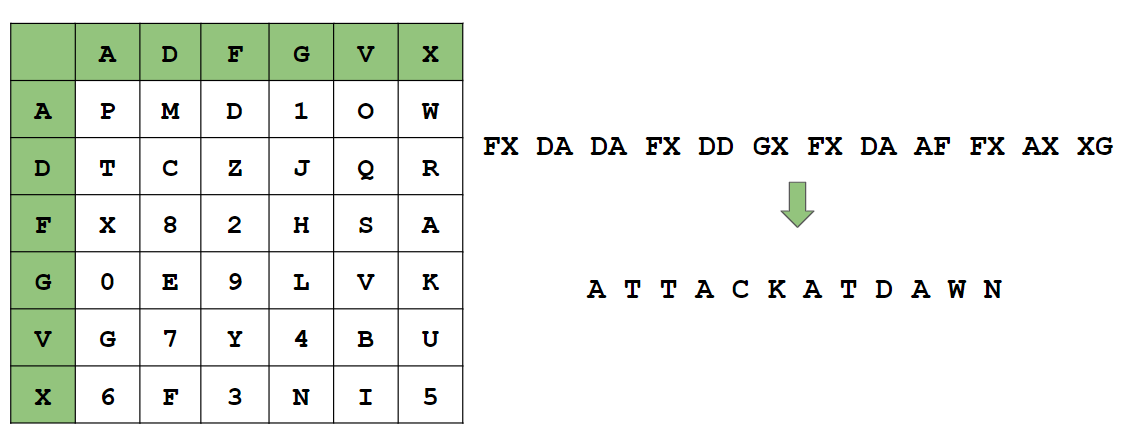
Step 2: Generate the transpositional matrix using the sorted key and filling the input characters of vertically (column-wise).



Step 3: Perform Columnar Transpose to match the key and extract the Polybius form.



Step 4: Use pair of characters as indices to lookup the Polybius square and extract the text and remove '-'.



**Test Cases:**

Test Cases provided in **qbox-2.zip** (Hackerrank zip format).

No. of Test Cases: **10** (excluding sample)

Encrypted Strings per test file: **8**, Total: 80

\*Use Challenge Slug **adfgvx** on hackerrank to add the challenge to any contest.