Data Structure and Algorithms

(HackerEarth solved Quiz) 2022

Aatish M. Patel

AIMDek Trainee

**Q 1) Monk and Rotation**

<https://www.hackerearth.com/problem/algorithm/monk-and-rotation-3-bcf1aefe/>

import java.util.\*;

class TestClass {

public static void printArray(int start,int end,int A[]){

for(int i=start;i<end;i++){

System.out.print(A[i]+" ");

}

}

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T!=0){

int N = sc.nextInt();

int K = sc.nextInt();

int A[] = new int[N];

for(int i=0;i<N;i++){

A[i] = sc.nextInt();

}

printArray((N - (K%N)),N,A);

printArray(0,(N - (K%N)),A);

System.out.println();

T-=1;

}

}

}

**Q 2) Monk and Inversions**

import java.util.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T!=0){

int N = sc.nextInt();

int M[][] = new int[N][N];

for(int i=0;i<N;i++){

for(int j=0;j<N;j++){

M[i][j]=sc.nextInt();

}

}

int ans = 0;

int temp = 0;

for(int i=0;i<N;i++){

for(int j=0;j<N;j++){

temp = M[i][j];

int a=i;

int b=j;

for(int x=0;x<N;x++){

for(int y=0;y<N;y++){

if(a <= x && b <= y){

if(temp > M[x][y]){

ans++;

}

}

}

}

}

}

System.out.println(ans);

T -= 1;

}

}

}

**Q 3) Cyclic shift**

T = int(input())

while T!=0:

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

s = input()

max = ""

p = -1;

for i in range(n):

if max < s:

max = s

d = i

elif max == s:

p = i-d

break

s = s[1:]+s[:1]

if p == -1:

print(d + (k-1)\*n)

else:

print(d + (k-1)\*p)

T-=1

**Q 4) Minimum AND xor OR**

import java.util.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T != 0){

int N = sc.nextInt();

int A[] = new int[N];

for(int i=0;i<N;i++){

A[i] = sc.nextInt();

}

Arrays.sort(A);

int min = A[0] ^ A[1];

for(int i=1;i<N-1;i++){

int temp = A[i] ^ A[i+1];

if(temp < min){

min = temp;

}

}

System.out.println(min);

T -= 1;

}

}

}

**Q 5) Monk and Nice Strings**

arr = []

n = int(input())

for \_ in range(n):

arr.append(input())

for i in range(n):

count = 0

for j in range(0,i,+1):

if arr[j] < arr[i]:

count += 1

print(count)

**Q 6) Monk and Suffix Sort**

line = input().split()

s = line[0]

k = int(line[1])

n = len(s)

arr = []

for i in range(n):

arr.append(s[i:])

arr.sort()

print(arr[k-1])

**Q 7) Monk being monitor**

import java.util.\*;

import java.lang.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T!=0){

int N = sc.nextInt();

int A[] = new int[N];

for(int i=0;i<N;i++){

A[i] = sc.nextInt();

}

Arrays.sort(A);

int res = 0;

int min = -1;

int current\_fre = 0;

for(int i=0;i<N;i++){

if(i!=N-1 && A[i]==A[i+1]){

current\_fre += 1;

}

else{

current\_fre += 1;

if(min == -1){

min = current\_fre;

}

else{

if(min>current\_fre){

min = current\_fre;

}

else{

res = Math.max(res,current\_fre-min);

}

current\_fre = 0;

}

}

}

if(res>0){

System.out.println(res);

}

else{

System.out.println(-1);

}

T -= 1;

}

}

}

**Q 8) Monk and Sorting Algorithm**

num = int(input())

A = list(map(int,input().strip().split()))

maxA = max(A)

mul = 1

rem = 10 \*\* 5

while maxA:

A.sort(key = lambda x: (x/mul)%rem)

print(' '.join(map(str,A)))

mul \*= rem

maxA //= rem

**Q 9) Wet clothes**

n,m,g = input().split()

t = list(map(int,input().split()))

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

Arr = []

for i in range(len(t)-1):

x=t[i+1]-t[i]

Arr.append(x)

m=max(Arr)

c=0

for i in range(len(a)):

if a[i]<=m:

c+=1

print(c)