**Data Structure**

**Data Structure and Algorithms**

**(HackerEarth solved Quiz) 2022**

**Sutariya Dipen C**

**AIMDek Trainee**

**Q 1) Monk and Rotation**

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

Python source code:

test\_Case = int(raw\_input())

while test\_Case!=0:

n,k = map(int,raw\_input().split())

arr = map(int,raw\_input().split())

index = n - (k%n)

for i in range(index, n):

print arr[i],

for i in range(index):

print arr[i],

print ""

test\_Case-=1

**Q 2) Monk and inversions**

[**https://www.hackerearth.com/problem/algorithm/monk-and-inversions-arrays-strings-e5aaa427/**](https://www.hackerearth.com/problem/algorithm/monk-and-inversions-arrays-strings-e5aaa427/)

**Python Souce Code**

test\_Case = int(raw\_input())

while test\_Case!=0:

n= int(raw\_input())

array=[]

for i in range(n):

array.append(map(int, raw\_input().split()))

count = 0

for i in range(n):

for j in range(n):

for v in range(i,n):

for k in range(j,n):

if array[i][j]> array[v][k]:

count+=1

print count

test\_Case -=1

**Q 3)Cyclic Shift**

<https://www.hackerearth.com/problem/algorithm/maximum-binary-number-cb9a58c1/>

**Python source Code**

test\_Case = int(raw\_input())

while test\_Case!=0:

n,k = map(int,raw\_input().split())

s = raw\_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)

print ""

test\_Case -= 1

**Q 4) Minimum And xor OR**

[**https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/minimum-and-xor-or-6a05bbd4/**](https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/minimum-and-xor-or-6a05bbd4/)

**Python Source Code**

test\_Case = int(raw\_input())

while test\_Case!=0:

n = int(raw\_input())

array = map(int, raw\_input().split(' '))

array.sort()

min = array[0] ^ array[1]

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

temp = array[i] ^ array[i+1]

if temp < min:

min = temp

print min

print ""

test\_Case -=1

**------------------------------------------------------------------------------------------------------------------**

**Sorting**

**Q 1)Monk and Nice String**

[**https://www.hackerearth.com/problem/algorithm/monk-and-nice-strings-3-e5800d05/**](https://www.hackerearth.com/problem/algorithm/monk-and-nice-strings-3-e5800d05/)

**Python Source Code**

n = input()

s =[]

for i in range(n):

s.append(raw\_input())

count = 0

for j in range(i):

if s[j] < s[i]:

count +=1

print (count)

**Q 2)Monk and Suffix Sort**

[**https://www.hackerearth.com/problem/algorithm/monk-and-suffix-sort-ebacdaf5/**](https://www.hackerearth.com/problem/algorithm/monk-and-suffix-sort-ebacdaf5/)

**Python Source Code**

s,n = map(str, raw\_input().split())

arr=[]

n=int(n)

for i in range(len(s)):

arr.append(s)

s = s[1:]

arr.sort()

print arr[n-1]

**Q 3)Monk and Sorting Algotihm**

[**https://www.hackerearth.com/problem/algorithm/monk-and-sorting-algorithm-3aa7826d/**](https://www.hackerearth.com/problem/algorithm/monk-and-sorting-algorithm-3aa7826d/)

**Python Source Code**

n = int(input())

arr = list(map(int,input().strip().split(" ")))

max\_arr = max(arr)

mul = 1

r = 10\*\*5

while max\_arr:

arr.sort(key = lambda x: (x/mul)%r)

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

mul \*= r

max\_arr //= r

**Q 4)Monk being monitor**

<https://www.hackerearth.com/problem/algorithm/monk-being-monitor-709e0fd3/>

**Python Source Code**

for \_ in range(input()):

n = input()

arr = map(int, raw\_input().split())

arr.sort()

res = 0

min = "-"

current\_freq = 0

for i in range(n):

if i!=n-1 and arr[i] == arr[i+1]:

current\_freq += 1

else:

current\_freq += 1

if min == "-":

min = current\_freq

else:

if min > current\_freq:

min = current\_freq

else:

res = max(res, current\_freq - min)

current\_freq = 0

if res>0:

print res

else:

print -1

**------------------------------------------------------------------------------------------------------------------**

**Searching**

**Q 1) Wet clothes**

[**https://www.hackerearth.com/problem/algorithm/wet-clothes-8a09a28e/**](https://www.hackerearth.com/problem/algorithm/wet-clothes-8a09a28e/)

**Python Source Code**

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)

**Q 2)Monk and Special Integer**

[**https://www.hackerearth.com/problem/algorithm/monk-and-special-integer-code-monk-e4b52aad/**](https://www.hackerearth.com/problem/algorithm/monk-and-special-integer-code-monk-e4b52aad/)

**Python Source Code**

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

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

arr = [0]

chk = 0

for i in range(n):

arr.append(arr[i] + a[i])

if a[i] > x:

chk = 1

if chk == 1:

print("-1")

elif arr[n] < x:

print(n)

else:

si = 1

li = n

while si <= li:

mid = (si+li)//2

chk = 0

j = mid

while j < n+1:

if arr[j] - arr[j-mid] > x:

break

j += 1

if j == n+1:

si = mid+1

else:

li = mid-1

print(si-1)

**Q 3) Monk and circular distance**

<https://www.hackerearth.com/problem/algorithm/circular-distance-3-c8592f16/>

**Python Source**

def search(arr,key):

si = 0

li = len(arr) - 1

while si < li:

mid = (si+li) // 2

if key >= arr[mid]:

si = mid + 1

else:

li = mid

return si

d = []

for \_ in range(input()):

x,y = map(int, raw\_input().split())

d.append(x\*x + y\*y)

d.sort()

for \_ in range(input()):

r = input()

r = r\*r

if r > d[len(d) - 1]:

print len(d)

else:

print search(d,r)