**Data Structure and Algorithms**

(HackerEarth solved Quiz) 2022

**Henil Patel**

**AIMDek Trainee**

**Arrays & Strings**

**Q 1) Monk and Rotation**

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

Python source code:

test = int(input())

for \_ in range(test):

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

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

x = k%n

print(\*(l[n-x:]+l[:n-x]))

**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 source code:

# Write your code here

test = int(input())

for \_ in range(test):

n= int(input())

matrix = []

for i in range(n):

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

matrix.append(a)

count=0

for i in range(n):

for j in range(n):

for p in range(n):

for q in range(n):

if i<=p and j<=q:

if matrix[i][j]>matrix[p][q]:

count+=1

print(count)

**Q 3) Cyclic shift**

[**https://www.hackerearth.com/problem/algorithm/maximum-binary-number-cb9a58c1/**](https://www.hackerearth.com/problem/algorithm/maximum-binary-number-cb9a58c1/)

Python source code:

test = int(raw\_input())

while test!=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 -= 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 = int(raw\_input())

while test!=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 -=1

**Sorting**

**Q 1) Monk and Nice Strings**

[**https://www.hackerearth.com/practice/algorithms/sorting/insertion-sort/practice-problems/algorithm/monk-and-nice-strings-3/**](https://www.hackerearth.com/practice/algorithms/sorting/insertion-sort/practice-problems/algorithm/monk-and-nice-strings-3/)

Python source code:

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 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())

n= int(n)

arr =[]

for i in range(len(s)):

arr.append(s)

s = s[1:]

arr.sort()

print(arr[n-1])

**Q 3) Monk being monitor**

[**https://www.hackerearth.com/problem/algorithm/monk-being-monitor-709e0fd3/**](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

**Q 4) Monk and Sorting Algorithm** [**https://www.hackerearth.com/practice/algorithms/sorting/insertion-sort/practice-problems/algorithm/monk-and-nice-strings-3/**](https://www.hackerearth.com/practice/algorithms/sorting/insertion-sort/practice-problems/algorithm/monk-and-nice-strings-3/)

Python source code:

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)

**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()))

gap =[]

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

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

gap.append(x)

m = max(gap)

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)