Data Structure and Algorithms

(HackerEarth solved Quiz) 2022

**Joshi Khushi**

AIMDek Trainee

**ARRAYS & STRINGS**

**Q 1) Monk and Rotation**

https://www.hackerearth.com/submission/68843288/

java source code:

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.\*;

class TestClass {

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

Scanner s = new Scanner(System.in);

int num = s.nextInt();

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

int len = s.nextInt();

int rotations = s.nextInt();

rotations = rotations % len;

s.nextLine();

String input = s.nextLine();

String[] inputArray = input.split(" ");

StringBuffer sb = new StringBuffer();

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

sb.append( inputArray[ (len + j - rotations) % len ] + " " );

}

System.out.print(sb);

System.out.println("");

}

}

}

**Q 2) Monk and Inversions**

https://www.hackerearth.com/submission/68865602/

java source code:

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.\*;

class TestClass {

public static void main (String[] args) {

Scanner s = new Scanner(System.in);

int t = s.nextInt();

while(t != 0) {

int n = s.nextInt();

int[][] a = new int[n][n];

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

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

a[i][j] = s.nextInt();

}

}

int inversions = 0;

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

for(int l = 0; l < n; l++) {

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

for(int k = 0; k <= l; k++) {

if(a[i][l] < a[j][k]) {

inversions++;

}

}

}

}

}

System.out.println(inversions);

t--;

}

}

}

**Q 3) Cyclic Shifts**

https://www.hackerearth.com/submission/68866644/

java source code:

import java.io.\*;

import java.util.\*;

class TestClass {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T-- > 0) {

int N; int K;

N = sc.nextInt();

K = sc.nextInt();

String input = sc.next();

String B = "";

String inter = input;

int d = 0;

int period = -1;

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

if (B.compareTo(inter) < 0) {

B = inter;

d = i;

}

else if (B.compareTo(inter) == 0) {

period = i - d;

break;

}

inter = inter.substring(1, inter.length()) + inter.substring(0, 1);

}

if(period == -1) {

System.out.println(d + (K - 1L ) \* N);

}

else {

System.out.println(d + ((K - 1L) \* period));

}

}

}

}

**Q 4) Minimum AND xor OR**

https://www.hackerearth.com/submission/68867204/

java source code:

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.\*;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

import java.util.StringTokenizer;

import java.util.\*;

class TestClass {

static class FastReader {

BufferedReader br;

StringTokenizer st;

public FastReader() {

br = new BufferedReader(

new InputStreamReader(System.in));

}

String next() {

while (st == null || !st.hasMoreElements()) {

try {

st = new StringTokenizer(br.readLine());

}

catch (IOException e) {

e.printStackTrace();

}

}

return st.nextToken();

}

int nextInt() { return Integer.parseInt(next()); }

long nextLong() { return Long.parseLong(next()); }

double nextDouble() { return Double.parseDouble(next()); }

String nextLine() {

String str = "";

try {

str = br.readLine();

}

catch (IOException e) {

e.printStackTrace();

}

return str;

}

}

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

FastReader fr = new FastReader();

int t = fr.nextInt();

while(t-- != 0) {

int n = fr.nextInt(),min = Integer.MAX\_VALUE ;

int arr[] = new int[n];

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

arr[i] = fr.nextInt();

}

Arrays.sort(arr);

for(int i = 0; i < n - 1; i++) {

int temp = (arr[i] & arr[i+1]) ^ (arr[i] | arr[i+1]);

if(temp < min)

min = temp;

}

System.out.println(min);

}

}

}

**SORTING**

**Q 5) Monk and Nice Strings**

https://www.hackerearth.com/submission/68867306/

java source code:

**import java.util.\*;**

**class TestClass {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**String[] s = new String[1010];**

**int j = 0;**

**s[0] = "A";**

**for(int i = 1 ; i <= n ; i++) {**

**String str = "";**

**str = sc.next();**

**for(j = i - 1 ; j >= 0 ; j--) {**

**if(s[j].compareTo(str) >= 0) {**

**s[j+1] = s[j];**

**}**

**else {**

**break;**

**}**

**}**

**System.out.println(j);**

**s[j+1] = str;**

**}**

**}**

**}**

**Q 6) Monk and Suffix Sort**

https://www.hackerearth.com/submission/68868969/

C++ source code:

**#include <bits/stdc++.h>**

**using namespace std;**

**#define mkp make\_pair**

**#define pb push\_back**

**#define scan(x) scanf("%d", &x)**

**#define ll long long int**

**#define MOD 1000000007**

**#define pii pair<int, int>**

**vector <string> v;**

**string a, b;**

**int main()**

**{**

**int k, i, j;**

**cin >> a >> k;**

**for (i = 0; a[i]; ++i) {**

**b= "";**

**for (j = i; a[j]; ++j) {**

**b += a[j];**

**}**

**v.pb(b);**

**}**

**sort(v.begin(), v.end());**

**cout << v[k-1] << endl;**

**return 0;**

**}**

**Q 7) Monk being monitor**

https://www.hackerearth.com/submission/68869189/

C++ source code:

**#include <bits/stdc++.h>**

**using namespace std;**

**#define MOD 1000000007**

**#define ll long long int**

**#define ld long double**

**#define pb push\_back**

**#define mkp make\_pair**

**#define pii pair<int, int>**

**#define pll pair<long long int, long long int>**

**#define sci(x) scanf("%d", &x)**

**#define scl(x) scanf("%lld", &x)**

**int mark[1000001];**

**set <pii > se;**

**int main()**

**{**

**int t, n, i, j, x, y;**

**cin >> t;**

**while (t--) {**

**memset(mark, 0, sizeof(mark));**

**cin >> n;**

**for (i = 0; i < n; ++i) {**

**sci(x);**

**mark[x]++;**

**}**

**x = y = -1;**

**for (i = 1; i <= 1000000; ++i) {**

**if (mark[i] && x == -1) {**

**x = i;**

**y = i;**

**}**

**if (mark[i] && (mark[i] > mark[x])) x = i;**

**if (mark[i] && (mark[i] < mark[y])) y = i;**

**}**

**//cout << x << " " << y << endl;**

**if (abs(mark[x] - mark[y]) > 0) cout << abs(mark[x]-mark[y]) << endl;**

**else cout << -1 << endl;**

**}**

**return 0;**

**}**