**ARRAYS**

1. **Rotate Game**

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class Array\_Rotation

{

static void RightRotate(int a[] , int n, int k)

{

k=k%n;

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

{

if(i<k)

{

System.out.print(a[n + i - k] + " ");

}

else

{

System.out.print(a[i - k]+ " ");

}

}

System.out.println();

}

public static void main(String args[])

{

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

int N=sc.nextInt();

int Array[] = new int[N]};

for(int index=0;index<N;index++)

{

arr[index]=sc.nextInt();

}

int K=sc.nextInt();

RightRotate(Array, N, K); }

}

1. **Good Pair**

#include <stdio.h>

int findPair(int nums[], int n, int target)

{ int count=0;

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

{

for (int j = i + 1; j < n; j++)

{

if (nums[i] + nums[j] == target) count++;

}

}

return count;

}

int main(void)

{

int N,K;

scanf("%d",&N);

int nums[N] ;

for(int index=0;index<N;index++)

scanf("%d",&nums[index]);

scanf("%d",&K);

int target = 7;

int n = sizeof(nums)/sizeof(nums[0]);

printf("%d ",findPair(nums, n, target));

return 0;

}

1. **Max and Min of an Array**

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in){

int N=sc.nextInt();

int []arr= new int[N];

for(int index=0;index<N;index++){ arr[index]=sc.nextInt();}

Arrays.sort(arr);

System.out.println(arr[0] +" " +arr[N-1]);

}

}

1. **Reverse the Array**

import java.util.\*;

public class Main

{

static void rvereseArray(int arr[],

int start, int end)

{

int temp;

while (start < end)

{

temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in){

int N=sc.nextInt();

int []arr= new int[N];

for(int index=0;index<N;index++){ arr[index]=sc.nextInt();}

rvereseArray(arr, 0, N);

for (int i = 0; i < size; i++)

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

}

}

1. **Search Element**

import java.util.\*;

public class Main

{

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in){

int N=sc.nextInt();

int []arr= new int[N];

for(int index=0;index<N;index++){ arr[index]=sc.nextInt();}

int K=sc.nextInt();

for(int index=0;index<N;index++){ arr[index]==K;{

System.out.println("1")

return ;}

System.out.println("0")

}

}

Homework  
**Little Ponny and Maximum Element**

#include <limits.h>

#include <stdio.h>

int arraySum(int arr[], int n)

{

int sum = 0;

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

sum += arr[i];

return sum;

}

int smallest(int arr[], int n)

{

int small = INT\_MAX;

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

if (arr[i] < small)

small = arr[i];

return small;

}

int minOp(int arr[], int n)

{

int sum = arraySum(arr, n);

int small = smallest(arr, n);

int minOperation = sum - (n \* small);

return minOperation;

}

int main()

{

int arr[] = { 5, 6, 2, 4, 3 };

int n = sizeof(arr) / sizeof(arr[0]);

printf("Minimum Operation = %d", minOp(arr, n));

return 0;

}

**Second Largest**

#include <stdio.h>

#include<stdlib.h>

int cmpfunc(const void\* a, const void\* b)

{

return (\*(int\*)a - \*(int\*)b);

}

void print2largest(int arr[], int arr\_size)

{

int i, first, second;

if (arr\_size < 2) {

printf(" Invalid Input ");

return;

}

qsort(arr, arr\_size, sizeof(int), cmpfunc);

for (i = arr\_size - 2; i >= 0; i--) {

if (arr[i] != arr[arr\_size - 1]) {

printf("The second largest element is %d\n",arr[i]);

return;

}

}

printf("There is no second largest element\n");

}

int main()

{

int arr[] = { 12, 35, 1, 10, 34, 1 };

int n = sizeof(arr) / sizeof(arr[0]);

print2largest(arr, n);

return 0;

}

**MINIMUM PICKS**

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int[] A=new int[n];

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

A[i]=sc.nextInt();

System.out.println(maxEvenMinOddDiff(A));

}

public static int maxEvenMinOddDiff(int[] A) {

int maxEven = Integer.MIN\_VALUE;

int minOdd = Integer.MAX\_VALUE;

for (int num : A) {

if (num % 2 == 0 && num > maxEven) {

maxEven = num;

} else if (num % 2 == 1 && num < minOdd) {

minOdd = num;

}

}

return maxEven - minOdd;

}

**Separate Odd Even**

import java.lang.\*;

import java.util.\*;

class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

while(t>0)

{

int a=sc.nextInt();

int arr[]=new int[a];

for(int i=0;i<a;i++)

{

arr[i]=sc.nextInt();

}

for(int i:arr)

if(i%2!=0)

System.out.print(i+" ");

System.out.println();

for(int i:arr)

if(i%2==0)

System.out.print(i+" ");

System.out.println();

t--;

}

}

}

**Multiple left rotations of the array**

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n1=sc.nextInt();

int[] A=new int[n1];

for(int i=0;i<n1;i++)

A[i]=sc.nextInt();

int n2=sc.nextInt();

int[] B=new int[n2];

for(int i=0;i<n2;i++)

B[i]=sc.nextInt();

int[][] result = rotateArray(A, B);

for (int i = 0; i < result.length; i++) {

for (int j = 0; j < result[i].length; j++) {

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

}

System.out.println();

}

}

public static int[][] rotateArray(int[] A, int[] B) {

int[][] result = new int[B.length][A.length];

for (int i = 0; i < B.length; i++) {

int rotations = B[i] % A.length;

for (int j = 0; j < A.length; j++) {

result[i][j] = A[(j + rotations) % A.length];

}

}

return result;

}

}

**Leaders in an array**

import java.lang.\*;

import java.util.\*;

class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int[] arr=new int[n];

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

arr[i]=sc.nextInt();

int l=arr.length-1;

List<Integer> list=new ArrayList<>();

for(int i=0;i<l;i++)

{

if (isGreater(arr, i, l) == true)

{

list.add(arr[i]);

}

}

list.add(arr[l]);

System.out.println(list);

}

private static boolean isGreater(int[] arr,int l,int r)

{

int max=arr[l];

boolean t=true;

for(int i=l+1;i<=r;i++)

{

if(max<arr[i])

t=false;

}

return t;

}

}

**Bulbs**

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int[] A=new int[n];

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

A[i]=sc.nextInt();

int count = 0,state=0;

for (int bulb : A) {

if (bulb == state) {

count++;

state = 1 - state;}

}

System.out.println(count);

}

}