**Youngest-oldest 1**

import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

if (!sc.hasNextInt()) return;

int n = sc.nextInt();

if (n <= 0) {

System.out.println("Invalid Input");

return;

}

int youngest = Integer.MAX\_VALUE;

int oldest = Integer.MIN\_VALUE;

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

if (!sc.hasNextInt()) {

System.out.println("Invalid Input");

return;

}

int age = sc.nextInt();

if (age < 0) {

System.out.println("Invalid Input");

return;

}

if (age < youngest) youngest = age;

if (age > oldest) oldest = age;

}

System.out.println("Youngest=" + youngest);

System.out.println("Oldest=" + oldest);

}

}

**Array 176**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int[] doors = new int[n];

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

doors[i] = sc.nextInt();

}

int search = sc.nextInt();

int index = -1;

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

if (doors[i] == search) {

index = i;

break;

}

}

if (index == -1) {

System.out.println("-1");

} else {

System.out.printf("Door Number is %03d-DN", index);

}

}

}

**Difference of the character 1**

import java.util.Scanner;

public class ValidStringCheck {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String S = sc.nextLine().trim();

int countStar = 0, countHash = 0;

for (char c : S.toCharArray()) {

if (c == '\*') countStar++;

else if (c == '#') countHash++;

}

int diff = countStar - countHash;

if (diff == 0) {

System.out.println("0");

} else {

int absDiff = Math.abs(diff);

if (absDiff % 2 == 1) {

System.out.printf("The Difference of the character in the given string: %03d%n", absDiff);

} else {

System.out.printf("The Difference of the character in the given string: 0%d%n", absDiff);

}

}

sc.close();

}

}

**Count the Positive and Negative Integer Number 1**

import java.util.Scanner;

public class CountPosNeg {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int posCount = 0, negCount = 0;

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

int num = sc.nextInt();

if (num > 0) {

posCount++;

} else if (num < 0) {

negCount++;

}

}

System.out.printf("Count of Positive Integer is %.2f%n", (double) posCount);

System.out.printf("Count of Negative Integer is %.2f%n", (double) negCount);

sc.close();

}

}

**ARRAY MEAN 4**

import java.util.Scanner;

public class ArrayMean {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt(); // size of array

int sum = 0;

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

sum += sc.nextInt();

}

double mean = (double) sum / n;

System.out.printf("Array Mean Value is %.2f%n", mean);

sc.close();

}

}

**Count distinct elements 8**

import java.util.\*;

public class DistinctElements {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

Set<Integer> distinct = new HashSet<>();

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

distinct.add(sc.nextInt());

}

int count = distinct.size();

if (count == 1) {

System.out.println("There are " + count + " distinct element in the array.");

} else {

System.out.println("There are " + count + " distinct elements in the array.");

}

sc.close();

}

}

**Search an Element 7**

import java.util.Scanner;

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 i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

int target = sc.nextInt();

boolean found = false;

for (int element : arr) {

if (element == target) {

found = true;

break;

}

}

if (found) {

System.out.println(target + " is presented in an array.");

} else {

System.out.println(target + " is not presented in an array.");

}

}

}

**Find K Largest Elements 1**

import java.util.Arrays;

import java.util.Scanner;

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 i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

int k = sc.nextInt();

Arrays.sort(arr);

System.out.print("The elements are in the order: ");

for (int num : arr) {

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

}

System.out.println();

System.out.print("The Kth value is " + k + " and Largest elements are ");

for (int i = n - 1; i >= n - k; i--) {

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

}

System.out.println();

}

}

**Sum of positive square 1**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int sum = 0;

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

int val = sc.nextInt();

if (val > 0) {

sum += val \* val;

}

}

System.out.println(sum);

}

}

**Second Array Max And Min 1**

import java.util.\*;

public class SecondLargeSmallSum {

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();

}

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

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

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

if (i % 2 == 0) {

evenPos.add(arr[i]);

} else {

oddPos.add(arr[i]);

}

}

Collections.sort(evenPos);

Collections.sort(oddPos);

int secondLargestEven = evenPos.get(evenPos.size() - 2);

int secondSmallestOdd = oddPos.get(1);

int sum = secondLargestEven + secondSmallestOdd;

System.out.println("Sum=" + sum);

sc.close();

}

}

**PATTERN**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

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

if(i==0 || i==n-1 || j==0||j==n-1){

System.out.print("\*");

}else{

System.out.print(" ");

}

}

System.out.println();

}

}

}