**YOUNGEST-OLDEST 1**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

if (n < 0) {

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

return;

}

int[] ages = new int[n];

boolean invalid = false;

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

ages[i] = scanner.nextInt();

if (ages[i] < 0) {

invalid = true;

}

}

if (invalid) {

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

return;

}

int youngest = ages[0];

int oldest = ages[0];

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

if (ages[i] < youngest) {

youngest = ages[i];

}

if (ages[i] > oldest) {

oldest = ages[i];

}

}

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;

// Count \* and #

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(); // array size

int[] arr = new int[n];

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

arr[i] = sc.nextInt(); // array elements

}

int target = sc.nextInt(); // element to search

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(); // size of the array

int sum = 0;

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

int val = sc.nextInt();

if (val > 0) {

sum += val \* val; // add square of positive numbers only

}

}

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(); // size of array

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<>();

// Separate elements by position

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

if (i % 2 == 0) {

evenPos.add(arr[i]);

} else {

oddPos.add(arr[i]);

}

}

// Sort both lists

Collections.sort(evenPos);

Collections.sort(oddPos);

// Find 2nd largest in even list

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

// Find 2nd smallest in odd list

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();

}

}

}