HACKERRANK QUESTIONS

DAY-4

1. import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

// Check if n is negative

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] = s.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);

}

}

2. import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Number of buildings

int n = sc.nextInt();

// Door numbers array

int[] doorNumbers = new int[n];

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

doorNumbers[i] = sc.nextInt();

}

// Friend's door number to search

int friendDoor = sc.nextInt();

// Search for friend's door number

int index = -1;

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

if (doorNumbers[i] == friendDoor) {

index = i;

break;

}

}

// Output the result

if (index == -1) {

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

} else {

// Format index with leading zeros (3 digits)

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

}

sc.close();

}

}

3. import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

int starCount = 0;

int hashCount = 0;

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

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

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

}

int diff = starCount - hashCount;

if (diff == 0) {

System.out.println(0);

} else if (diff > 0) {

if (diff == 1) {

// Special case for difference == 1

System.out.println("The Difference of the character in the given string: 001");

} else {

// For other positive differences ( > 1 ), 2-digit zero padded

System.out.printf("The Difference of the character in the given string: %02d\n", diff);

}

} else {

// For negative difference: negative sign + 3-digit zero padded absolute value

System.out.printf("The Difference of the character in the given string: -%03d\n", Math.abs(diff));

}

sc.close();

}

}

4. import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt(); // number of elements

int positiveCount = 0;

int negativeCount = 0;

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

int num = sc.nextInt();

if (num > 0) positiveCount++;

else if (num < 0) negativeCount++;

// zero is neither positive nor negative, so we skip counting it

}

// Print with 2 decimal places (as float)

System.out.printf("Count of Positive Integer is %.2f\n", (float) positiveCount);

System.out.printf("Count of Negative Integer is %.2f\n", (float) negativeCount);

sc.close();

}

}

5. import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

double sum = 0;

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

int num = sc.nextInt();

sum += num;

}

double mean = sum / n;

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

sc.close();

}

}

6. import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Read the size of the array

int n = sc.nextInt();

// Use a set to store distinct elements

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

// Read array elements and add to the set

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

int element = sc.nextInt();

distinctElements.add(element);

}

// Number of distinct elements

int count = distinctElements.size();

// Print the result

System.out.println("There are " + count + " distinct element" + (count > 1 ? "s" : "") + " in the array.");

sc.close();

}

}