Here is a **list of very simple questions** covering all **five main topics** for your test.

**1. Arrays (Easy Level)**

1. Find the largest element in an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

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

{

arr [i] = sc.nextInt();

}

int max = Integer.MIN\_VALUE;

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

{

if(arr[i]>max)

{

max = arr[i];

}

}

System.out.println(max);

}

}

1. Find the smallest element in an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

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

{

arr [i] = sc.nextInt();

}

int min = Integer.MAX\_VALUE;

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

{

if(arr[i]<min)

{

min = arr[i];

}

}

System.out.println(min);

}

}

1. Find the sum of all elements in an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

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

{

arr [i] = sc.nextInt();

}

int sum =0;

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

{

sum += arr[i];

}

System.out.println(sum);

}

}

1. Find the average of elements in an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

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

{

arr [i] = sc.nextInt();

}

int sum =0;

double avg=0;

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

{

sum += arr[i];

}

System.out.println("the sum of the array is: "+sum);

avg = (double)sum/a;

System.out.println("the average of the sum is: "+avg);

}

}

1. Reverse an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

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

{

arr [i] = sc.nextInt();

}

for(int i=a-1;i>=0;i--)

{

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

}

}

}

1. Count how many times a number appears in an array.

import java.util.\*;

public class queue

{

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

int a = sc.nextInt();

int arr [] = new int [a];

int tar = 2;

int c=0;

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

{

arr [i] = sc.nextInt();

}

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

{

if(arr[i]==tar)

c++;

}

System.out.println("the number 2 appears "+c+" times");

}

}

1. Find the second largest element in an array.

import java.util.\*;

public class SecondLargest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int arr[] = new int[a];

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

arr[i] = sc.nextInt();

}

int max = Integer.MIN\_VALUE;

int secondMax = Integer.MIN\_VALUE;

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

{

if (arr[i] > max)

{

secondMax = max;

max = arr[i];

}

else if (arr[i] > secondMax && arr[i] != max)

{

secondMax = arr[i];

}

}

System.out.println("Max value is "+max);

if (secondMax == Integer.MIN\_VALUE) {

System.out.println("No second largest element found.");

} else {

System.out.println("The second largest element is: " + secondMax);

}

sc.close();

}

}

1. Find the second smallest element in an array.

import java.util.\*;

public class SecondLargest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int arr[] = new int[a];

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

arr[i] = sc.nextInt();

}

int min = Integer.MAX\_VALUE;

int secMin = Integer.MAX\_VALUE;

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

{

if (arr[i] < min)

{

secMin = min;

min = arr[i];

}

else if (arr[i] < secMin && arr[i] != min)

{

secMin = arr[i];

}

}

System.out.println("min value is "+min);

if (secMin == Integer.MAX\_VALUE) {

System.out.println("No second largest element found.");

} else {

System.out.println("The second min element is: " + secMin);

}

sc.close();

}

}

1. Find the difference between the largest and smallest elements in an array.

import java.util.\*;

public class SecondLargest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int arr[] = new int[a];

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

arr[i] = sc.nextInt();

}

int large = Integer.MIN\_VALUE;

int Min = Integer.MAX\_VALUE;

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

{

if (arr[i] > large)

{

large = arr[i];

}

if (arr[i] < Min)

{

Min = arr[i];

}

}

int dif = large - Min ;

System.out.println("large is: " + large);

System.out.println("min value is "+Min);

System.out.println("diff is "+dif);

sc.close();

}

}

1. Check if an array is sorted in ascending order.

import java.util.\*;

public class CheckSorted {

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();

}

boolean isSorted = true;

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

if (arr[i] > arr[i + 1]) {

isSorted = false;

break;

}

}

if (isSorted) {

System.out.println("The array is sorted in ascending order.");

} else {

System.out.println("The array is NOT sorted.");

}

sc.close();

}

}

1. bubble sort

import java.util.\*;

public class SwapFirstLast {

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();

}

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

{

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

{

if(arr[j]>arr[j+1])

{

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1]=temp;

}

}

}

System.out.println("Sorted array "+Arrays.toString(arr));

}

}

12.Swap the first and last elements of an array.

import java.util.\*;

public class SwapFirstLast {

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();

}

if (n > 1)

{

int temp = arr[0];

arr[0] = arr[n - 1];

arr[n - 1] = temp;

}

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

{

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

}

System.out.println("");

System.out.println("Array after swapping first and last elements: " + Arrays.toString(arr));

sc.close();

}

}

13. Find the missing number in an array of numbers from 1 to N.

import java.util.\*;

public class CheckSorted {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int arr[] = new int[n-1];

int sum=0;

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

arr[i] = sc.nextInt();

}

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

{

sum+=arr[i];

}

int expectedSum = (n\*(n+1))/2;

int missing = expectedSum - sum ;

System.out.println("Missing number is "+missing);

}

}

**2. Strings (Easy Level)**

16. Find the length of a string.

import java.util.\*;

public class SwapFirstLast {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

int length = 0;

char arr [] = s.toCharArray();

int n = arr.length;

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

{

length++;

}

System.out.println(length);

}

}

1. Convert a string to uppercase.

import java.util.\*;

public class SwapFirstLast {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

char arr [] = s.toCharArray();

int n = arr.length;

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

{

if(arr[i]>='a'&& arr[i]<='z')

{

arr[i] = (char)(arr[i]-32);

}

}

String result = new String (arr);

System.out.println(result);

}

}

1. Convert a string to lowercase.

import java.util.\*;

public class SwapFirstLast {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

char arr [] = s.toCharArray();

int n = arr.length;

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

{

if(arr[i]>='A'&& arr[i]<='Z')

{

arr[i] = (char)(arr[i]+32);

}

}

String result = new String (arr);

System.out.println(result);

}

}

1. Count the number of vowels and consonants in a string.

import java.util.\*;

public class VowelsAndConsonants {

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in);

String str = sc.nextLine();

int vowels = 0, consonants = 0;

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

{

char ch = str.charAt(i);

if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z') {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

vowels++;

} else {

consonants++;

}

}

}

System.out.println("Vowels: " + vowels + ", Consonants: " + consonants);

}

}

1. Check if a string is a palindrome (same forward and backward).

import java.util.\*;

public class PalindromeCheck {

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

String s = sc.nextLine();

boolean pal = true;

for(int i=0,j=s.length()-1;i<j;i++,j--)

{

if(s.charAt(i)!=s.charAt(j))

{

pal = false;

}

}

if(pal)

{

System.out.println("pal");

}

else{

System.out.println("not pal");

}

}

}

1. Reverse a string without using extra space.

import java.util.\*;

public class PalindromeCheck

{

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in);

String s = sc.nextLine();

char arr [] = s.toCharArray();

int start =0;

int end = s.length() -1;

while(start<end)

{

char temp = arr[start];

arr[start]=arr[end];

arr[end]=temp;

start++;

end--;

}

String result = new String(arr);

System.out.println(result);

}

}

1. Remove all spaces from a string.

public class RemoveSpaces {

public static void main(String[] args) {

String str = "Hello World";

String result = "";

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

if (str.charAt(i) != ' ') {

result += str.charAt(i);

}

}

System.out.println("String without spaces: " + result);

}

}

23. Find the first repeated character in a string.

24. Find the first non-repeating character in a string.

25. Count the number of words in a sentence.

public class FirstRepeatedCharacter {

public static void main(String[] args) {

String str = "hello all";

char arr [] = str.toCharArray();

int n = arr.length;

int c=1;

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

{

if(arr[i]==' ')

{

c++;

}

}

System.out.println(c);

}

}

1. Replace all occurrences of a character in a string.

public class ReplaceCharacter {

public static void main(String[] args) {

String str = "Hello World";

char target = 'o';

char replacement = '0';

String result = "";

char arr [] = str.toCharArray();

int n = arr.length;

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

if (arr[i] == target) {

result = result + replacement; //hell +0 = hell0

} else

{

result += arr[i];

}

}

System.out.println("String after replacement: " + result);

}

}

1. Check if two strings are anagrams (contain the same letters in any order).

import java.util.\*;

public class ReplaceCharacter {

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in); //hello olleh

String s1 = sc.nextLine();

String s2 = sc.nextLine();

char arr1 [] = s1.toCharArray();

char arr2 [] = s2.toCharArray();

int n=arr1.length; //5

int m=arr2.length;

int c=0;

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

{

for(int j=0;j<m;j++)

{

if(arr1[i]==arr2[j])

{

c++;

break;

}

}

}

if(c==n)

{

System.out.println("Anagram");

}

else

{

System.out.println("NO");

}

}

}

1. Concatenate two strings without using built-in functions.

import java.util.\*;

public class ReplaceCharacter {

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in); //hello olleh

String s1 = sc.nextLine();

String s2 = sc.nextLine();

String result = "";

for(int i=0;i<s1.length();i++)

{

result = result + s1.charAt(i);

}

for(int j=0;j<s2.length();j++)

{

result = result + s2.charAt(j);

}

System.out.println(result);

}

}

1. Extract the first three characters of a string.

public class ExtractFirstThreeCharacters {

public static void main(String[] args) {

String str = "Hello";

String result = "";

for (int i = 0; i < 3 && i < str.length(); i++) {

result += str.charAt(i);

}

System.out.println("First three characters: " + result);

}

}

1. Check if a string contains only digits.

public class ExtractFirstThreeCharacters

{

public static void main(String[] args) {

String str = "0123456789";

boolean digits = true;

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

{

if(str.charAt(i)<'0'||str.charAt(i)>'9')

{

digits = false;

}

}

if(digits)

{

System.out.println("Yes");

}

else

{

System.out.println("No");

}

}

}

**3. Linked List (Easy Level)**

31. Insert an element at the beginning of a linked list.

32. Insert an element at the end of a linked list.

33. Insert an element at a given position in a linked list.

34. Delete the first element of a linked list.

35. Delete the last element of a linked list.

36. Find the length of a linked list.

37. Find the middle element of a linked list.

38. Reverse a linked list.

39. Check if a linked list is empty.

40. Search for an element in a linked list.

**4. Basic Recursion (Easy Level)**

41. Print numbers from 1 to N using recursion.

42. Print numbers from N to 1 using recursion.

43. Find the sum of numbers from 1 to N using recursion.

44. Find the factorial of a number using recursion.

45. Find the nth Fibonacci number using recursion.

46. Find the sum of digits of a number using recursion.

47. Reverse a string using recursion.

48. Find the power of a number (x^n) using recursion.

49. Count the number of digits in a number using recursion.

50. Find the greatest common divisor (GCD) of two numbers using recursion.

**5. Medium-Level Recursion (For Your Test)**

51. Check if a given string is palindromic using recursion.

52. Solve the Tower of Hanoi problem for N disks.

53. Implement binary search using recursion.

54. Generate all subsets of a given string using recursion.

55. Find all permutations of a given string using recursion.

These questions cover **everything you need** for your test in **arrays, strings, linked lists, and recursion**.

Would you like help solving any of these?