**1. How do you get a list of all employee names and ages?**List<String> namesAndAges = employeeList.stream().map(emp -> emp.getName() + " - " + emp.getAge()).toList();  
  
**2. How do you get a list of gender and city of all employees?**  
List<String> genderCityList = employeeList.stream() .map(emp -> emp.getGender() + " - " + emp.getCity()).toList();  
  
**3. How to get unique department names?**List<String> uniqueDepartments = employeeList.stream().map(Employee::getDeptName).distinct().toList();  
  
**4. How to get unique city names?**List<String> uniqueCities = employeeList.stream().map(Employee::getCity).distinct().toList();  
  
**5. How to find the oldest employee?**Optional<Employee>oldestEmp=employeeList.stream().max(Comparator.comparingInt(Employee::getAge));  
  
**6. How to find the average age of employees?**Double avgAge = employeeList.stream().collect(Collectors.averagingInt(Employee::getAge));  
  
**7. How to get all employees whose name starts with 'A'?**List<Employee> t=empList.stream().filter(e->e.getName().startsWith(“A”)).toList();

**8. How to sort employees by age in descending order?**List<Employee> sortedByAgeDesc=employeeList.stream().sorted(Comparator.comparing(Employee::getAge).reversed()).toList();  
  
**9. How to group employee names by department?**Map<String,List<String>> namesByDept=employeeList.stream().collect(Collectors.groupingBy(Employee::getDeptName,Collectors.mapping(Employee::getName, Collectors.toList())));  
  
10. **How to calculate total age of all employees?**int totalAge = employeeList.stream().mapToInt(Employee::getAge).sum();

**11. How to partition employees based on age > 30?**Map<Boolean,List<Employee>> partitioned=employeeList.stream().collect(Collectors.partitioningBy(emp -> emp.getAge() > 30));  
  
**12. How to find all active employees with salary > 5000?**List<Employee> highEarners = employeeList.stream().filter(emp -> emp.isActiveEmp() && emp.getSalary() > 5000).toList();  
  
**13. How to get names of employees from the "IT" department?**List<Employee> a=empList.stream().filter(e->”IT”.equals(emp.getDeptName()))  
.map(Employee::getName).toList();  
  
**14. How to count active vs inactive employees?**Map<Boolean,Long> activeCount=employeeList.stream().collect(Collectors.groupingBy(Employee::isActiveEmp, Collectors.counting()));  
  
**15. How to get employee names from "Mumbai" sorted by salary descending?**List<String> mumbaiHighEarners = employeeList.stream().filter(emp -> "Mumbai".equals(emp.getCity())) .sorted(Comparator.comparingDouble(Employee::getSalary).reversed())  
.map(Employee::getName)  
.toList();  
  
**16. How to get all employees older than 25?**List<Employee> ageAbove25 = employeeList.stream().filter(e->e.getAge()>25).toList();  
  
**17. How to count employees in each department?**Map<String,Long> empCountByDept=employeeList.stream().collect(Collectors.groupingBy(Employee::getDeptName, Collectors.counting()));  
  
**18. How to get average age of employees in each department?**Map<String,Double> avgAgeByDept=employeeList.stream().collect(Collectors.groupingBy(Employee::getDeptName, Collectors.averagingInt(Employee::getAge)));  
  
**19. How to get a list of distinct employee ages?**List<Integer> distinctAges = employeeList.stream().map(Employee::getAge).distinct().toList();

**20. How to get the salary of each employee?**List<Double> salaryList = employeeList.stream().map(Employee::getSalary).toList();  
  
**21. How to get average salary per department?**Map<String,Double>avgSalaryByDept=employeeList.stream().collect(Collectors.groupingBy(Employee::getDeptName, Collectors.averagingDouble(Employee::getSalary)));  
  
**22. Find the first repeatating character in the string using stream api?**String str=”Java code threads”;  
LinkedHashMap<Character,Long> input=str.toLowerCase().chars().mapToObj(c->(char)c).  
collect(Collectors.groupingBy(x->x,LinkedhashMap::new,Collectors.counting()));  
Character character=input.entrySet().stream().filter(x->x.getValue()>1).  
map(x->x.getKey()).findFirst().get();  
  
**22. Find the first non-repeatating character in the string using stream api?**LinkedHashMap<Character,Long> output=str.toLowerCase().chars().mapToObj(c->(char)c).  
Collect(Collectors.groupingBy(x->x,LinkedHashMap::new,Collectors.counting()));

Character character=input.entrySet().stream().filter(x->x.getValue()==1).  
map(x->x.getKey()).findFirst().get();

**23. Find the most repeated character in string**String str=”jashdjgsgxsafc”;  
Optional<Character> input=str.chars().mapToObj(c->(char)c).collect(Collectors.groupingBy(  
Function.identity(),Collectors.counting()).  
entrySet().  
stream().  
max(Map.Entry.<Character,Long>comparingByValue()).  
map(Map.Entry::getKey);  
  
input.ifPresentOrElse(c->Syestem.out.println(“Most repeated charcter”+c),  
()->System.out.println(“String is empty or has no repeated characters”));  
  
**24. Find the frequency of each character in string**String str=”Ravichavan”;  
Map<Character,Long> input=str.chars().mapToObj(c->(char)c).collect(Collectors.groupingBy(  
Function.identity().Collectors.counting()));  
  
**25. Find the frequency of String in a List**List<String > sList=Arrays.asList(“Ravi”,”Sumit”,”Karun”,”Nava”,”Nava”,”Ravi”);  
Map<String,Long> input=sList.stream().collect(Collectors.groupingBy(Function.identity(),  
Collectors.counting()));

**26. Find the longest word in a string**  
String str=”Virat Kohali and mahendrasingh dhoni”;  
String longestWord=Arrays.stream(str.split(“ ”)).max(Comparator.comparingInt(  
String::length)).orElse(“ “);  
  
**26. Find the smallest word in a string**  
String smallestWord=Arrays.stream(str.split(“ “)).min(Comparator.comparingInt(  
String::length)).orElse(“ ”);  
  
**27. Find the Second longest word in the string**  
String secondLongest=Arrays.stream(str.split(“ ”)).sorted(Comparator.comparing(  
String::length).reversed()).skip(1).findFirst().orElse(“ ”);  
  
**28. Find the 2nd highest word length**  
int secondLength=Arrays.stream(str.split(“ “)).map(x->x.length()).sorted(Copmarator.  
reverseOrder()).skip(1).findFirst().get();  
  
**29. Find the Duplicate character in string**  
String str=”jhdasgcdvcgahv”;  
List<Character> list=str.chars().distinct().mapToObj(c->(char)c).collect(Collectors.toList);  
  
**30. How do you count by each and every word length**  
List<String> str=Arrays.asList(“helloworld”);  
Map<Integer,Long> input=str.stream().collect(Collectors.groupingBy(String::length,  
Collectors.counting));  
  
**31. How do you find the groupLength**  
List<String> str=Arrays.asList(“banana”,”apple”,”chicku”);  
Map<Integer,Long> input=str.stream().collect(Collectors.groupingBy(String::length));  
  
**32. Given a list of Strings, find the frequency of each word using Java Streams**  
List<String> words = Arrays.asList("apple", "cherry", "apple", "orange", "banana", "cherry");  
Map<String,Long> input=Arrays.stream().collect(Collectors.groupingBy(Function.identity(),  
Collectors.counting()));  
  
**33. Find the Second highest value in the list**List<Integer> list=Arrays.asList(1,2,3,7,9,4,6);  
Optional<Integer> input=list.stream().sorted(Comparator.reverseOrder()).limit(2).skip(1).  
findFirst();  
  
  
**34. Find the Second lowest value in the list**  
Optional<Integer> input=list.stream().sorted(Comparator.naturalOrder()).limit(2).skip(1).  
findFirst();  
  
**35. Find the max value in the list**int max=list.stream().max(Integer::compareTo).get();  
  
**36. Find the min value in the list**  
int min=list.stream().min(Integer::compareTO).get();  
  
**37. How do you find the unique element**List<Integer> list=Arrays.asList(1,2,3,4,5,3,2);  
List<Integer> h=list.stream().distinct().toList();  
  
Set<Integer> f=new HashSet<>();  
List<Integer> n=list.stream().filter(k->!f.add(k)).toList();  
  
**38. How do we find in the list even number**  
List<Integer> list=Arrays.asList(1,23,4,5,8,9);  
List<Integer> input=list.stream().filter(n->n%2==0).toList();  
  
**39. Pair anagrams from a list of strings. one word consider  only one anagram**  
String [] a1={"pat","tap","pan","map","team","tree","meat"};  
List<String> list=Arrays.asList(a1);  
Map<Object,List<String>> list1=list.stream().stream(Collectors.groupingBy(  
x->toLowerCase().split(“ ”)).sorted(Collectors.toList()));  
  
**40. Find the sum of all the elements in a list.**List<Integer> list=Arrays.asList(1,2,3,4,5);  
int ad=list.stream().mapToInt(Integer::intValue).sum();  
  
**41. Sort a list of strings in alphabetical order**   
List<String> list=Arrays.asList(“Virat”,”Ravi”,”Balu”,”Pavan”);  
List<String> num=list.stream().sorted().toList();  
  
  
  
  
  
  
  
**42. find the 2nd highest occurring character in a string using Java 8 Stream API**  
String st = "thetimeisthreeoclock";  
Optional<Character> secondMax = st.chars()  
            .mapToObj(c -> (char) c)  
            .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()))  
            .entrySet()  
            .stream()  
            .sorted(Map.Entry.<Character, Long>comparingByValue().reversed())   
       .skip(1) // skip the first (most frequent)  
         .map(Map.Entry::getKey)  
          .findFirst(); // get the second highest  
   secondMax.ifPresentOrElse(

            c -> System.out.println("Second most repeated character: " + c),  
         () -> System.out.println("No second most frequent character found")

        );

**42. How do we reverse String**  
String s=”Ravi”;  
String rev=””;  
for(int i=s.length()-1; i>=0; i--)  
{  
 rev=rev+s.charAt(i);

}  
System.out.println(rev);  
  
  
**43. How do we check the String is palindrome or not**  
String s=”Dipak”;  
String rev=””;  
for(int i=s.length()-1;i>=0; i--)  
{  
 rev=rev+s.charAt(i);  
}  
if(s.equals(rev))  
{  
 Sytem.out.println(“Palindrome number”);  
}  
else {  
 System.out.println(“ Palindrome String”); }  
**43. Resverse String**public class ReverseStringWord {  
public static void main(String[] args) {  
 String str="Satish is good man";  
 String[] s=str.split(" ");  
 String rev="";  
 for(int i=s.length-1;i>=0;i--) {  
 rev=rev+s[i]+" ";  
 }System.*out*.println(rev);  
 }  
}

**44. Swap String multiple ways**  
package com.string;  
public class SwapString {  
 public static void main(String[] args) {  
 String s1="Hello";  
 String s2="word";  
 String temp=s1;  
 s1=s2;  
 s2=temp;  
 System.*out*.println(s1);  
 System.*out*.println(s2);  
  
 // second ways  
 s1=s1+s2;  
 s2=s1.substring(0,s1.length()-s2.length());  
 s1=s1.substring(s2.length());  
 System.*out*.println(s1);  
 System.*out*.println(s2);  
  
 // Using array  
 String[] arr = {"Hello", "word"};  
 String temp1 = arr[0];  
 arr[0] = arr[1];  
 arr[1] = temp1;  
 System.*out*.println("s1 = " + arr[0]); // word  
 System.*out*.println("s2 = " + arr[1]); // Hello  
}  
}

**45. String UpperCase**  
public class StringUpperCase {  
 public static void main(String[] args) {  
 String str="Bhagwan hai kaha re tu";  
 String upper="";  
 String lower="";  
 int count=0;  
 for(int i=0;i<str.length();i++) {  
 char ch=str.charAt(i);  
 if(ch>=65 && ch<=90) {  
 upper+=ch;  
 count++;  
 }else {  
 lower+=ch;  
 }  
 }  
 System.*out*.println(upper);  
 System.*out*.println(lower);  
 System.*out*.println(count);  
 }  
}  
  
**46. String LowerCase**  
public class StringLowerCase {  
 public static void main(String[] args) {  
 String str="Ai khuda hai kaha re tu";  
 String lower="";  
 int count=0;  
 for(int i=0;i<str.length();i++) {  
 char ch=str.charAt(i);  
 if(ch>='a' && ch<='z') {  
 lower+=ch;  
 count++;  
 }  
 }  
 System.*out*.println(lower);  
 System.*out*.println(count);  
 }  
}

**47. Find Vowels**public class FindTheVowels {  
 public static void main(String[] args) {  
 String str="Kadmo me pe tere badal jhukenge jab tak tujhe aihsas hai";  
 String vowels="";  
 String consonant="";  
 int count =0;  
 for(int i=0;i<str.length();i++) {  
 char ch=str.charAt(i);  
 if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u') {  
 vowels+=ch;  
 count++;  
 }  
 else if(ch!='a' || ch!='e' || ch!='i' || ch!='o' || ch!='u'){  
 consonant+=ch;  
 }  
 }System.*out*.println(vowels);  
 System.*out*.println(consonant);  
 System.*out*.println(count);  
 }  
}  
  
**48. Count First word**  
public class CountFirstWordLength {  
 public static void main(String[] args) {  
 String str=" Navanath can run 100 km";  
 System.out.println(lengthOfFirstWord(str));  
 }  
 static int lengthOfFirstWord(String s) {  
 String str=s.trim();  
 int count=0;  
 for(int i=0;i<str.length();i++) {  
 if(str.charAt(i)!=' ') {  
 count++;  
 }else  
 break;  
 }  
 return count;  
 }  
}

**49. Count Last word**public class CountFirstWordLength {  
 public static void main(String[] args) {  
 String str=" Navanath can run 100 km";  
 System.*out*.println(*lengthOfFirstWord*(str));  
}  
 static int lengthOfFirstWord(String s) {  
 String str=s.trim();  
 int count=0;  
 for(int i=0;i<str.length();i++) {  
 if(str.charAt(i)!=' ') {  
 count++;  
 }else  
 break;  
 }  
 return count;  
 }  
}  
  
**50. Capitalize First and Last character**  
public class CapitalizeFirstLastChar {  
 public static void main(String[] args) {  
 String s = " Satish is good man";  
 String[] words=s.trim().split("\\s+");  
 StringBuilder result=new StringBuilder();  
 for(String word:words) {  
 int len=word.length();  
 if(len==1) {  
 result.append(Character.*toUpperCase*(word.charAt(0)));  
 }else {  
 result.append(Character.*toUpperCase*(word.charAt(0)))  
 .append(word.substring(1,len-1))  
 .append(Character.*toUpperCase*(word.charAt(len-1)));  
 }  
 result.append(" ");  
 }  
 System.*out*.println(result.toString().trim());

}  
}

**51. Capitalize First Charcter**public class CapitalizeFirstLetter {  
 public static void main(String[] args) {  
 String s = " Satish is good man";  
 String[] words = s.trim().split("\\s+");  
 StringBuilder result = new StringBuilder();  
 for (String word : words) {  
 int len = word.length();  
 if (len == 1) {  
 result.append(Character.*toUpperCase*(word.charAt(0)));  
 } else {  
 result.append(Character.*toUpperCase*(word.charAt(0))) // First letter  
 .append(word.substring(1)); // Rest as it is  
 }  
 result.append(" ");  
 }  
 System.*out*.println(result.toString().trim());  
 }  
}  
 **52. Capitalize Last Character**public class CapitalizeLastLetter {  
 public static void main(String[] args) {  
 String s = " Satish is good man";  
 String[] words = s.trim().split("\\s+"); // split by one or more spaces  
 StringBuilder result=new StringBuilder();  
 for(String word:words) {  
 int len=word.length();  
 if(len==1) {  
 result.append(Character.*toUpperCase*(word.charAt(0)));  
 } else {  
 result.append(word.substring(0, len - 1)) // all except last  
 .append(Character.*toUpperCase*(word.charAt(len - 1)));   
 }  
 result.append(" ");  
 }  
 System.*out*.println(result.toString().trim());

}  
}

**44. Find the prime number**int no=7;  
int temp=0;  
for(int i=2;i<=no-1;i++){  
 if(no%i==0){  
 temp=temp+1;  
 }  
}  
if(temp==0){  
 System.out.println(no+” is prime” );  
}  
else  
 System.out.println(no+”is not prime”); **44. Find the longest without repeating String**  
String input =”abcabbcc”;  
int maxLength=0;  
int startIndex=0;  
  
for (int i=0; i<input.length();i++)  
{  
 Set<Character> seen=new HashSet<>();  
 int j=i;  
  
 while(j<input.length() && !seen.contains(input.charAt(j)))  
{  
 seen.add(input.charAt(j));  
 j++;  
}  
 if(j-i>maxLength)  
 {  
 maxLength=j-i;  
 startIndex=i;  
 }  
}  
  
String result = input.substring(startIndex, startIndex + maxLength);  
System.out.println("Longest substring without repeating characters: " + result);

**46. Smallest substring in the string**  
public static String minWindow(String s, String t)  
{   
 int minLength = Integer.MAX\_VALUE;  
 String result=””;  
   
 for(int i=0; i<=s.length() – t.length(); i++)  
 {  
 for(int j=i+t.length(); j<=s.length();j++)  
 {  
 String sub=s.substring(i,j);  
 if(containsAll(sub, t) {  
 if (sub.length() < minLength) {  
 minLength=sub.length();  
 result =sub;  
 }  
 }  
 }  
}  
return result;  
}  
public static boolean containsAll(String window, String target){  
 for(char c : target.toCharArray()){  
 if(window.indexOf(c)== -1){  
 return false;  
 }  
 window = window.replaceFirst(c+ “ ”, replacement: “ “);  
 }  
 return true;  
}  
public static void main(String[]args){  
 String s=”ADOBECODEBANC”;  
 String t=”BANC”;  
 System.out.println(minWindow(s,t));  
}

**47. Find the max value from the array**

int a[]={0,1,4,2,7,9};

int max=0;

for(int i=1;i<a.length;i++){

if(a[i]>max){

max=a[i];

}

}  
 System.out.println("Maximum element is: "+max);  
  
**48. Find the min value from the array**  
  
int a[]={1,2,6,4,9,0,5};  
int min=a[0];  
for(int i=0;i<a.length;i++){  
 if(a[i]<min){  
 min=a[i];  
}  
}  
System.out.println(min);  
  
  
**49. Find big string from array**  
String a[] = {"Ravi", "Nava", "Ashish"};  
String max=a[0];  
for(int i=0;i<a.length;i++){  
 if(a[i].length()>max.length()){  
 max=a[i];  
 }  
}  
System.out.println(max);  
  
  
  
  
**50. Find the small string**  
String a[] = {"Ravi", "Nava", "Ashish"};  
String min=a[0];  
for(int i=0;i<a.length;i++){  
 if(a[i].length()<min.length()){  
 min=a[i];  
 }  
}  
System.out.println(min);  
  
**51. Find Duplicate element in array**  
int a[]={19,27,3,19,2,4,6,9};  
for(int i=0;i<a.length;i++){  
 for(int j=i+1; j<a.length;j++){  
 if(a[i]==a[j] && i!=j){  
 System.out.println(a[j]);   
 }  
 }  
}  
  
**51. Find multiple duplicate element**  
  
int[] a = {2, 3, 4, 2, 3, 6};  
Set<Integer> s = new HashSet<>();  
 for (int no : a) {  
 if (!s.add(no)) {  
 System.out.println(no + " ");  
 }  
 }  
  
**52. Find first duplicate number**  
int[] a = {2, 3, 4, 2, 3, 6};  
int temp=0;  
for(int i=0;i<a.length-1;i++){  
 for(int j=i+1;j<a.length;j++){  
 if(a[i]==a[j] && i!=j){  
 System.out.println(a[i]);  
 temp=temp+1;  
 break;  
 }  
 }  
if(temp>0){  
 break;  
 }  
}  
  
**53. Find the second largest element**  
int a[]={10,23,45,67,54};  
int temp=0;  
for (int i = 0; i < a.length; i++) {  
 for (int j = i + 1; j < a.length; j++) {  
 if (a[i] < a[j]) {  
 temp = a[i];  
 a[i] = a[j];  
 a[j] = temp;  
 }  
 }  
 }   
System.out.println("Second largest: " + a[1]);  
  
**54. Find the second smallest element**  
int a[]={10,23,45,67,54};  
int temp=0;  
for(int i=0;i<a.length;i++){  
 for(int j=i+1;j<a.length;j++){  
 if(a[i]>a[j]){  
 temp=a[i];  
 a[i]=a[j];  
 a[j]=temp;  
 }  
 }  
}  
System.out.println(“Second smallest ”+a[1]);

**55. find the palindrome number**  
int no = 121;  
int temp = no;  
int rev = 0, rem;  
while (temp != 0) {  
 rem = temp % 10;  
 rev = rev \* 10 + rem;  
 temp = temp / 10;  
 }

if (no == rev) {  
 System.out.println("Palindrome");

} else {  
 System.out.println("Not");

}  
  
  
**56. Fibonacci series**  
int a=0, b=1;  
System.out.println(a+” ”+b);  
for(int i=0; i<=10; i++){  
 c=a+b;  
 System.out.println(“”+c);  
 a=b;  
 b=c;  
}  
  
**57. Factorial number**  
int no=5;  
int fact=1;  
for(int i=1; i<=no; i++){  
 fact=fact\*i;  
}  
System.out.println(fact);  
  
  
  
  
  
  
**58. Find the UpperCase and Lowercase in the String**String str = "Welcome to the automation";  
String lower = "";  
String upper = "";  
 for (int i = 0; i < str.length(); i++) {  
 char ch = str.charAt(i);  
 if (ch >= 65 && ch <= 90) {  
 upper = upper + ch;  
 } else {  
 lower = lower + ch;  
 }  
 }  
 System.out.println("Lowercase and other characters: " + lower);  
 System.out.println("Uppercase letters: " + upper);  
 }  
  
**59. To count the number of uppercase and lowercase letters in a string.**String str=”Welcome to the Automation”;  
int upper=0;  
int lower=0;  
  
for(int i=0;i<str.length;i++){  
 char ch=str.charAt(i);  
 if (ch >= 'A' && ch <= 'Z') {  
 upper++;  
 } else if (ch >= 'a' && ch <= 'z') {  
 lower++;  
 }}  
System.out.println(lower);  
System.out.println(upper);  
  
  
  
  
  
  
  
  
  
  
  
**60. find the vowels in a string using Java**  
String str="Welcome to the Automation";  
String vowels = "";  
for(int i=0;i<str.length();i++){  
 char ch=str.charAt(i);  
 if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'){  
 vowels+=ch;  
 }  
}  
System.out.println("Vowels in the string: " + vowels);  
  
**61. Count the vowels in the string using java**String str=”Welcome to the automation”;  
String vowels=””;  
int count=0;  
for(int i=0;i<str.length();i++){  
 char ch=str.charAt(i);  
 if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'){  
 vowels+=ch;  
 count++;  
 }  
System.out.println("Vowels in the string: " + vowels);  
System.out.println("Total number of vowels: " + count);  
  
**62. Combine into one string and split into single words**  
String str = "virat, rohit, rinku, manoj";  
String s[] = str.split(","); // split by comma and space  
 for (String name : s) {  
 System.out.println(name);  
}  
  
**63. Java program to reverse words in string?**  
String str= "ravi is here";  
String s=str.split(“ ”);  
String rev=””;  
for(int i=s.length-1; i>=0; i--){  
 rev=rev+s[i]+””;  
}  
System.out.println(rev);

64. stream use configure object you need emp id, name, age dept emp sort between different employee same department  
  
 Map<String, List<Employee>> result = employeeList.stream()

            .collect(Collectors.groupingBy(

                Employee::getDeptName,

                Collectors.collectingAndThen(

                    Collectors.toList(),

                    empList -> empList.stream()

                        .sorted(Comparator.comparingInt(Employee::getAge))

                        .collect(Collectors.toList())

                )

            ));

        // Print result

        result.forEach((dept, empList) -> {

            System.out.println("Department: " + dept);

            empList.forEach(System.out::println);

        });  
  
**Max rearrange number**  
  
public class MaxMinReaarange {

public static void main(String[] args) {

int[] arr= {1,2,3,4,5,6,7,8};  
 int start =0;  
 int end=arr.length-1;  
 List<Integer> result=new ArrayList<>();

while(start<=end) {  
 if(start!=end) {  
 result.add(arr[end]);  
 result.add(arr[start]);  
 }else {  
 result.add(arr[start]);  
 }  
 start++;  
 end--;  
 }  
 System.out.println(result);  
 }  
}  
 **Missing Number in Array**  
**public** **class** MissingArray {  
 **public** **static** **void** main(String[] args) {  
 **int**[] a = {1,2,3,4,5,6,7,9}; // missing 8  
 **int** n = a.length + 1; // total numbers should be 9  
 **int** expectedSum = n \* (n + 1) / 2; // sum of 1...9  
 **int** actualSum = 0;  
 **for** (**int** i = 0; i < a.length; i++) {  
 actualSum += a[i];  
 }  
 **int** missingNumber = expectedSum - actualSum;  
 System.***out***.println("Missing number is: " + missingNumber);

}

}  
  
**Sum Element  
public** **class** SumElement {  
 **public** **static** **void** main(String[] args) {  
 **int**[] arr = {1, 2, 3, 4, 5};  
 **int** sum=0;  
 **for**(**int** i=0;i<arr.length;i++) {  
 sum+=arr[i];  
 }System.***out***.println("sum= "+sum);  
 }

**Parenthesis**public class BalancedParentheses {  
 public static boolean isBalanced(String str) {  
 Stack<Character> stack=new Stack();  
 for(int i=0;i<str.length();i++) {  
 char ch=str.charAt(i);  
 if(ch=='(' || ch=='{' || ch=='[' ) {  
 stack.push(ch);  
 }else if(ch==')' || ch=='}' || ch==']') {  
 if(stack.isEmpty())  
 return false;  
   
 char top=stack.pop();

if ((ch == '}' && top != '{') ||

(ch == ']' && top != '[') ||

(ch == ')' && top != '(')) {

return false;

}  
 }  
 }  
 return stack.isEmpty();  
 }  
public static void main(String[] args) {  
 String input1 = "{[()]}";  
 String input2 = "{[(])}";  
 System.*out*.println(input1 + " -> " + (*isBalanced*(input1) ? "Balanced" : "Not Balanced"));

System.*out*.println(input2 + " -> " + (*isBalanced*(input2) ? "Balanced" : "Not Balanced"));

**2**public class BracketPairCounter {  
 public static int countBracketPairs(String str) {  
 int open = 0;  
 int pairCount = 0;  
 for (char ch : str.toCharArray()) {  
 if (ch == '(') {  
 open++;  
 } else if (ch == ')') {  
 if (open > 0) {  
 pairCount++;  
 open--;  
 }  
 }  
 }

return pairCount;  
 }

public static void main(String[] args) {  
 String str = "(((()()()()()((((()(())))";  
 int result = *countBracketPairs*(str);  
 System.*out*.println("Total valid () pairs: " + result);

}

}

3  
**public** **class** CurlyBracketPairs {  
 **public** **static** **int** countValidCurlyPairs(String s) {  
 **int** open = 0, pairs = 0;  
 **for** (**char** ch : s.toCharArray()) {  
 **if** (ch == '{') {  
 open++;

} **else** **if** (ch == '}') {

**if** (open > 0) {

pairs++;

open--;

}

}

}

**return** pairs;

}

**public** **static** **void** main(String[] args) {

String s = "}{{}}{{{";

**int** result = *countValidCurlyPairs*(s);

System.***out***.println("Valid {} pairs: " + result);

}

}