https://www.linkedin.com/in/kunalkr19

Java Coding Questions with Example

Index

Array

Q. Swap two numbers using temporary variable

Input - a = 10, b = 20;
Output - a = 20 , b = 10

```
public class A {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        int temp = a;
        a = b;
        b = temp;
        System.out.println(a);
        System.out.println(b);
    }
}
```

Q. Swap two numbers without temporary variable

Input - a = 10, b = 20;
Output - a = 20 , b = 10

```
public class A {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        a = a + b;
        b = a - b;
        a = a - b;
        System.out.println(a);
        System.out.println(b);
    }
}
```

Q. Check a number is even or odd

```
public class A {
    public static void main(String[] args) {
        int number = 15;
        if (number%2==0){
            System.out.println("number is even");
        }else {
            System.out.println("number is odd");
        }
    }
}
```

Q. Find largest of three numbers

```
Input -  int a = 10;
         int b = 15;
         int c = 5;
Output - 15
```

```
public class A {
    public static void main(String[] args) {
        int a = 10;
        int b = 15;
        int c = 5;
        int largest = a > (b>c?b:c)?a:(b>c?b:c);
        System.out.println(largest);
    }
}
```

Q. Check a year is leap year or not

```
public class A {
    public static void main(String[] args) {
        int year = 2020;
        if ((year%4==0)||((year%4==0)&&(year%100!=0))){
            System.out.println("leap year");
        }else {
            System.out.println("not leap year");
        }
    }
}
```

Q. Find factorial of a number using for loop

```
public class A {
    public static void main(String[] args) {
        int number = 5;
        int fact = 1;
        for (int i=2;i<=5;i++){
            fact = fact*i;
        }
        System.out.println(fact);
    }
}
```

Q. Fibonacci Series

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

```
public class A {
    public static void main(String[] args) {
        int n1 = 0;
        int n2 = 1;
        int count = 10;
        System.out.print(n1+" "+n2);
        for (int i=2;i<count;i++){
            int n3 = n1 + n2;
            System.out.print(" "+n3);
            n1 = n2;
            n2 = n3;
        }
    }
}
```

Q. Check a number is prime or not

```
public class A {
    public static void main(String[] args) {
        int number = 4;
        if (isPrime(number)){
            System.out.println("number is prime");
        }else {
            System.out.println("number is  not prime");
        }
    }

    private static boolean isPrime(int number) {
        if (number<=1){
            return false;
        }
        for (int i=2;i<Math.sqrt(number);i++){
            if (number%i==2){
                return false;
            }
        }
        return true;
    }
}
```

Q. Search an element in an array

Input - int[] arr = {1,2,3,6,8,9,5,0};
Output - Element is present at index 3

```
public class A {
    public static void main(String[] args) {
        int[] arr = {1,2,3,6,8,9,5,0};
        int element = 6;
        for (int i=0;i<arr.length;i++){
            if (arr[i]==element){
                System.out.println("Element is present at index "+i);
                break;
            }
        }
    }
}
```

Q. Sort an Array

Input - {3,4,6,7,3,6,2};
Output - [2, 3, 3, 4, 6, 6, 7]

```
public class A {
    public static void main(String[] args) {
        int [] arr = {3,4,6,7,3,6,2};
        Arrays.sort(arr);
        System.out.println(Arrays.toString(arr));
    }
}
```

Q. Find largest element in an array

Input - {10,324,45,90,9898};
Output - 9898

```
public class A {
    public static void main(String[] args) {
        int[] arr = {10,324,45,90,9898};
        int largest = arr[0];
        for (int i=1;i<arr.length;i++){
            if (arr[i]>largest){
                largest = arr[i];
            }
        }
        System.out.println(largest);
    }
}
```

Q. Find minimum element in an array

Input - {10,324,45,90,9898};
Output - 10

```
public class A {
    public static void main(String[] args) {
        int[] arr = {10,324,45,90,9898};
        int minimum = arr[0];
        for (int i=1;i<arr.length;i++){
            if (arr[i]<minimum){
                minimum = arr[i];
            }
        }
        System.out.println(minimum);
    }
}
```

Q. Merge two Arrays

```
input -  int a[] = {30,25,40};
        int b[] = {45,50,55,60,65};
Output - 30 25 40 45 50 55 60 65
```

```
public class A {
    public static void main(String[] args) {
        int a[] = {30,25,40};
        int b[] = {45,50,55,60,65};
        int length = a.length + b.length;
        int[] c = new int[length];
        for (int i=0;i<a.length;i++){
            c[i] = a[i];
        }
        for (int i =0;i<b.length;i++){
            c[a.length+i] = b[i];
        }
        for (int x:c){
            System.out.print(x+" ");
        }
    }
}
```

String

Q. How to take an input String

```
public class A{
    public static void main(String[] args) {
        System.out.println("Enter a String :");
        Scanner scan = new Scanner(System.in);
        String str = scan.next();
        System.out.println(str);
    }
}
```

Q. To get a character from a String

```
public class A {
    public static void main(String[] args) {
        String str = "javaProgramming";
        int index = 4;
        System.out.println(str.charAt(index));
    }
}
```

Q. Replace a character at a specific index in a String

Input - javaDrogramming

Output - javaProgramming

```
public class A {
    public static void main(String[] args) {
       String str = "javaDrogramming";
       int index = 4;
       char ch = 'P';
       str = str.substring(0,index) + ch + str.substring(index+1);
       System.out.println(str);
    }
}
```

Q. Replace a character at a specific index in a String using StringBuilder

Input - javaDrogramming

Output - javaProgramming

```
public class A {
    public static void main(String[] args) {
       String str = "javaDrogramming";
       int index = 4;
       char ch = 'P';
       StringBuilder str1 = new StringBuilder(str);
       str1.setCharAt(index,ch);
        System.out.println(str1);
    }
}
```

Q. Replace a character at a specific index in a String using StringBuffer

Input - javaDrogramming
Output - javaProgramming

```
public class A {
    public static void main(String[] args) {
       String str = "javaDrogramming";
       int index = 4;
       char ch = 'P';
       StringBuffer str1 = new StringBuffer(str);
       str1.setCharAt(index,ch);
        System.out.println(str1);
    }
}
```

Q. Reverse a String

Input - ashutosh
Output - hsotuhsa

```
public class A {
    public static void main(String[] args) {
       String str = "java";
       String revStr = "";
       for (int i=0;i<str.length();i++){
           revStr =   str.charAt(i)+revStr;
       }
        System.out.println(revStr);
    }
}
```

Q. To sort a String

Input - hello
Output - ehllo

```
public class A {
    public static void main(String[] args) {
       String str = "hello";
       char[] charArr = str.toCharArray();
        Arrays.sort(charArr);
        String sortedString = new String(charArr);
        System.out.println(sortedString);
    }
}
```

Q. Swapping pair of characters in a String

Input - computer
Output - [o, c, p, m, t, u, r, e]

```
public class A {
    public static void main(String[] args) {
       String str = "computer";
       swapPair(str);
    }

    private static void swapPair(String str) {
        if (str==null||str.isEmpty()){
            System.out.println("String is null or Empty");
        }
        char[] ch = str.toCharArray();
        for (int i=0;i<str.length()-1;i=i+2){
            char temp = ch[i];
            ch[i] = ch[i+1];
            ch[i+1] = temp;
        }
        System.out.println(ch);
    }
}
```

Q. Find a unicode value of character

```
public class A {
    public static void main(String[] args) {
       String str = "abcxyzABCXYZ";
        System.out.println(str.codePointAt(0));
        System.out.println(str.codePointAt(1));
        System.out.println(str.codePointAt(2));
        System.out.println(str.codePointAt(3));
        System.out.println(str.codePointAt(4));
        System.out.println(str.codePointAt(5));
    }
}
```

Output -

```
97
98
99
120
121
122
```

Q. Remove Leading zeros from a String

Input - 0000abc
Output - abc

```
public class A {
    public static void main(String[] args) {
       String str = "0000abc";
       removeZero(str);
    }
    private static void removeZero(String str) {
        int i=0;
        while (i<str.length()&& str.charAt(i)=='0') {
            i++;
        }
        StringBuffer sb = new StringBuffer(str);
        sb.replace(0,i,"");
        System.out.println(sb.toString());
    }
}
```

Q. Compare to String

```
public class A {
    public static void main(String[] args) {
      String str1 = "testing";
      String str2 = "testing";
      if (str1.equals(str2)){
          System.out.println("Both Strings are equals");
      }else {
          System.out.println("String are not equals");
      }
    }}
```

Q. Check String is palindrome or not

```
public class A {
    public static void main(String[] args) {
      String str = "madam";
      if (checkPalindrome(str)){
          System.out.println("String is palindrome");
      }else {
          System.out.println("String is not palindorme");
      }
    }

    private static boolean checkPalindrome(String str) {

        int left = 0;
        int right = str.length()-1;
        while (left<right){
            if (str.charAt(left)!=str.charAt(right)){
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

Q. Java Program to count occurrence of each character in a String

Input - ashutosh
Output - {a=1, s=2, t=1, u=1, h=2, o=1}

```
public class A {
    public static void main(String[] args) {
        String str = "ashutosh";
        Map<Character, Integer> charMapCount = new
HashMap<>();
        for (Character ch :str.toCharArray()){
            if (charMapCount.containsKey(ch)){
                charMapCount.put(ch,charMapCount.get(ch)+1);
            }else {
                charMapCount.put(ch,1);
            }
        }
        System.out.println(charMapCount);
    }
}
```

Q. Reverse a String using Recursion

Input - ashutosh
Output - hsotuhsa

```
public class A {
    public static void main(String[] args) {
        String str = "java";
        String revStr = reverseString(str);
        System.out.println(revStr);
    }

    private static String reverseString(String str) {
        if (str==null||str.length()<=1){
            return str;
        }
        return reverseString(str.substring(1))+str.charAt(0);
    }
}
```

Q. Count number of words in a String

Input - java programming questions
Output - 3

```
public class A {
    public static void main(String[] args) {
        String str = "java programming questions";
        System.out.println(countWord(str));
    }
    private static int countWord(String str) {
        int wordCount = 1;
        for (int i=0;i<str.length();i++){
            if (str.charAt(i)==' ' && i<str.length()-1 && str.charAt(i+1)!=' '){
                wordCount++;
            }
        }
        return wordCount;
    }
}
```

Q. Find Duplicate character in a String

Input - programming
Output - r g m

```
public class A {
    public static void main(String[] args) {
        String str = "programming";
        duplicateCharacter(str);
    }

    private static void duplicateCharacter(String str) {
        Map<Character,Integer> charMapCount = new
HashMap<>();
        for (Character ch:str.toCharArray()){
            if (charMapCount.containsKey(ch)){
                charMapCount.put(ch,charMapCount.get(ch)+1);
            }else {
                charMapCount.put(ch,1);
            }
        }
        charMapCount.forEach((key,value)->{
            if (value>1){
                System.out.print(key+” “);
            }
        });
    }
}
```

Q. Reverse a String using stack

Input - ashutosh
Output - hsotuhsa

```
public class A {
    public static void main(String[] args) {
        String str = "java";
        Stack<Character> stack = new Stack<>();
        for (int i=0;i<str.length();i++){
            stack.push(str.charAt(i));
        }
        System.out.println("Reverse of String :");
        while (!stack.empty()){
            System.out.print(stack.pop());
        }
    }
}
```

Q. Find first non-repeating character in a String

Input - java
Output - j,v

```
public class A {
    public static void main(String[] args) {
        String str = "java";
        Map<Character,Integer> charMapCount = new
HashMap<>();
        for (Character ch:str.toCharArray()){
            if (charMapCount.containsKey(ch)){
                charMapCount.put(ch,charMapCount.get(ch)+1);
            }else {
                charMapCount.put(ch,1);
            }
        }
        for (int i=0;i<str.length();i++){
            char c = str.charAt(i);
            if (charMapCount.get(c)==1){
                System.out.println("first non repeating character "+c);
                break;
            }
        }
    }
}
```

Q. Find the longest common prefix

Input - {"cat", "cable","camera"}
Output - ca

```
public class A {
    public static void main(String[] args) {
        String str[] = {"cat", "cable","camera"};
        String result = findLongestPrefix(str);
        System.out.println(result);
    }

    private static String findLongestPrefix(String[] str) {
        if (str==null||str.length==0){
            return "";
        }
        String lcp = str[0];
        for (int i=1;i<str.length;i++){
            String currentWord = str[i];
            int j=0;

while(j<currentWord.length()&&j<lcp.length()&&currentWord.charAt(j)==lcp.charAt(j)){
                j++;
            }
            if (j==0){
                return "";
            }
            lcp = currentWord.substring(0,j);
        }
        return lcp;
    }
}
```

Q. Check for anagram

Input - String str1 = "car";
            String str2 = "rac";
Output - Strings are anagram

```
public class A {
    public static void main(String[] args) {
        String str1 = "car";
        String str2 = "rac";
        if (checkAnagram(str1,str2)){
            System.out.println("String are anagram");
        }else {
            System.out.println("String are not anagram");
        }
    }

    private static boolean checkAnagram(String str1, String str2) {
        if (str1.length()!=str2.length()){
            return false;
        }
        int[] countArr = new int[26];
        for (int i=0;i<str1.length();i++){
            countArr[str1.charAt(i)-'a']++;
            countArr[str2.charAt(i)-'a']--;
        }
        for (int i=0;i<countArr.length;i++){
            if (countArr[i]!=0){
                return false;
            }
        }
        return true;
    }
}
```

Stream API

Q. Sort a given list in reverse order

Input - Arrays.*asList*(12, 2, 4, 5, 2, 4, 8);
Output - [12, 8, 5, 4, 4, 2, 2]

```
public class A {
   public static void main(String[] args) {
           List<Integer> list = Arrays.asList(12, 2,
4, 5, 2, 4, 8);
           List<Integer> newList = list.stream().
                   sorted(Comparator.reverseOrder()).
                   collect(Collectors.toList());
           System.out.println(newList);

   }
}
```

Q. Given a list of strings, write a Java 8 program to join the strings with '[' as a prefix, ']' as a suffix, and ',' as a delimiter.

Input - Arrays.*asList*("adam","mike","sam");
Output - [adam],[mike],[sam]

```
public class A {
  public static void main(String[] args) {
       List<String>list=Arrays.asList("adam","mike","sam");
       String result = list.stream()
               .map(s -> "[" + s + "]")
               .collect(Collectors.joining(","));

       System.out.println(result);

   }
}
```

Q. Find the maximum and minimum of a list of integers

Input - Arrays.*asList*(1,4,6,8,2);

Output - 8,1

```
public class A {
   public static void main(String[] args) {

       List<Integer>list=Arrays.asList(1,4,6,8,2);
       int max =Collections.max(list);
       int min =Collections.min(list);
       System.out.println(max);
       System.out.println(min);

   }
}
```

Q. Merge two unsorted arrays into a single sorted array using Java 8 streams

Input - int[] array1 = {5, 3, 9, 1};
    int[] array2 = {7, 2, 8, 4};
Output - [1, 2, 3, 4, 5, 7, 8, 9]

```
public class A {
   public static void main(String[] args) {

       int[] array1 = {5, 3, 9, 1};
       int[] array2 = {7, 2, 8, 4};

       int[] mergeArrya =
IntStream.concat(Arrays.stream(array1),
Arrays.stream(array2))
               .sorted().distinct().toArray();
       System.out.println(Arrays.toString(mergeArrya));
   }
}
```

Q. Get the three maximum and three minimum numbers from a given list of integers

```
public class A {
   public static void main(String[] args) {


List<Integer>list=Arrays.asList(10,50,40,62,4,1,3,5,9);
       list.stream().sorted(Comparator.reverseOrder())
               .limit(3).forEach(System.out::println);
       list.stream().sorted().
               limit(3).forEach(System.out::println);
   }
}
```

Q.check if two strings are anagrams or not using Java 8 streams

```
public class A {
   public static void main(String[] args) {
           String s1="listen";
           String s2="silent";
           String join1 =
Arrays.stream(s1.split("")).sorted().collect(Collectors.join
ing(""));
           String join2 =
Arrays.stream(s2.split("")).sorted().collect(Collectors.join
ing(""));
           if(join1.equals(join2)) {
               System.out.println("anagram");
           }else {
               System.out.println("not anagram");
           }
   }
}
```

Q. Sort a list of strings according to the increasing order of their length

Input - `Arrays.asList("BBB","A","CCC","DDDD");`
Output - [A, BBB, CCC, DDDD]

```
public class A {
   public static void main(String[] args) {


List<String>list=Arrays.asList("BBB","A","CCC","DDDD");
           List<String> list2 = list.stream()

.sorted(Comparator.comparingInt(String::length))
                   .collect(Collectors.toList());
           System.out.println(list2);
   }
}
```

Q Find the common elements between two arrays

Input - `int[]a1= {10,20,30,40};`
        `int[]a2= {40,30,50};`
Output - [40, 30]

```
public class A {
   public static void main(String[] args) {
       int[]a1= {10,20,30,40};
       int[]a2= {40,30,50};
```

```
        Set<Integer> set =
Arrays.stream(a1).boxed().collect(Collectors.toSet());
        int[] commonElement = Arrays.stream(a2).
                filter(a->set.contains(a)).toArray();
        System.out.println(Arrays.toString(commonElement));
    }
}
```

Q. Reverse each word of a string using Java 8 streams

Input - `Arrays.asList("mike","adam","vikas");`
Output - [ekim, mada, sakiv]

```
A public class A {
    public static void main(String[] args) {
        List<String>list=Arrays.asList("mike","adam","vikas");
        List<StringBuffer> collect = list.stream()
                .map(word->new StringBuffer(word).reverse())
                .collect(Collectors.toList());
        System.out.println(collect.toString());
    }
}
```

Q. Find the sum of the first 10 natural numbers

```
public class A {
    public static void main(String[] args) {
        int sum = IntStream.rangeClosed(1, 10).sum();
        System.out.println(sum);

    }
}
```

Q. Reverse an integer array

Input - {15,20,5,60,70,25,30,45,96};
Output - reverse array [96, 45, 30, 25, 70, 60, 5, 20, 15]

```
public class A {
   public static void main(String[] args) {
       int[] num={15,20,5,60,70,25,30,45,96};
       System.out.println("original array
"+Arrays.toString(num));

       int[] reverseArray = IntStream.rangeClosed(1,
num.length).map(i -> num[num.length - i]).toArray();
       System.out.println("reverse array
"+Arrays.toString(reverseArray));


   }
}
```

Q. Print the first 10 even numbers

```
public class A {
   public static void main(String[] args) {
       IntStream.rangeClosed(1,20).filter(i->i%2==0).
               forEach(System.out::println);
   }
}
```

Q. Find the most repeated element in an array

Input - {1,2,3,4,5,6,3,4,6,7,3,5};

```
Output - 3
```

```
public class A {
   public static void main(String[] args) {
       int[] array = {1,2,3,4,5,6,3,4,6,7,3,5};
       Map<Integer, Long> collect =
Arrays.stream(array).boxed().

collect(Collectors.groupingBy(Function.identity(),
Collectors.counting()));

       Integer key = Collections.max(collect.entrySet(),
Map.Entry.comparingByValue()).getKey();
       System.out.println(" most repeated element in an
array+"+key);

   }
}
```

Q. Check if a string is a palindrome using Java 8 streams

```
public class A {
   public static void main(String[] args) {
       String str="nitin";
       String str2="";
       for (int i = str.length()-1; i>=0; i--) {
           str2+=str.charAt(i);
       }
       if(str2.equals(str)){
           System.out.println("string is palindrome");
       }else{
           System.out.println("string is not palindrome");
       }
   }
}
```

****

Q. Extract duplicate elements from an array

```
public class A {
   public static void main(String[] args) {

       int []num={5,25,10,36,25,45,65,75,45,85,95,5};
       Map<Integer, Long> collect =
Arrays.stream(num).boxed().

collect(Collectors.groupingBy(Function.identity(),
Collectors.counting()));


collect.entrySet().stream().filter(str->str.getValue()>1)
               .forEach(entry->
System.out.println(entry.getKey()+entry.getValue()));

   }
}
```

Q. Find the first repeated character in a string

```
public class A {
   public static void main(String[] args) {

       String str = "banana";
       Map<Character, Long> charCountMap = str.chars()
```

```
               .mapToObj(c -> (char)
c).collect(Collectors.groupingBy(c -> c, LinkedHashMap::new,
Collectors.counting()));
       Character c = charCountMap.entrySet()
               .stream().filter(entry -> entry.getValue() >
1)

.map(Map.Entry::getKey).findFirst().orElseThrow(() -> new
IllegalArgumentException("No non-repeated character found in
the string."));
       System.out.println(c);

   }
}
```

Q.Find the first non-repeated character in a string

```
public class A {
   public static void main(String[] args) {

       String str = "banana";
       Map<Character, Long> charCountMap = str.chars()
               .mapToObj(c -> (char)
c).collect(Collectors.groupingBy(c -> c, LinkedHashMap::new,
Collectors.counting()));
       Character c = charCountMap.entrySet()
               .stream().filter(entry -> entry.getValue() ==
1)

.map(Map.Entry::getKey).findFirst().orElseThrow(() -> new
IllegalArgumentException("No non-repeated character found in
the string."));
       System.out.println(c);

   }
}
```

Q.Print the first 10 odd numbers

```
public class A {
   public static void main(String[] args) {

       IntStream.rangeClosed(1, 20).filter(i -> i % 2 != 0)
               .forEach(System.out::println);
   }
}
```

Q Write a Java 8 program to get the last element of an array.

Original array: [15, 2, 65, 85, 74, 36, 74, 52, 25, 36, 74, 85]
Last element: 85

```
 public class A {
   public static void main(String[] args) {

       int[] num = {15, 2, 65, 85, 74, 36, 74, 52, 25, 36,
74, 85};
       System.out.println("Original array: " +
Arrays.toString(num));
       int lastElement = Arrays.stream(num)
               .reduce((first, second) -> second)
.orElseThrow(() -> new IllegalArgumentException("Array is
empty"));
       System.out.println("Last element: " + lastElement);
   }
}
```

Q. Write a program to append char in char

input- {A, B, C}
output->[A_X, B_Y, C_Z]

```
public class A {
   public static void main(String[] args) {

       Stream<Character> charStream = Stream.of('A', 'B',
'C');
       charStream.forEach(ch -> {
           char newChar = (char) (ch + 23);
           System.out.println(ch + "_" + newChar);
       });
   }
}
```

Q.How to find duplicate elements in a given integers list in java using Stream functions?

```
public class A {
   public static void main(String[] args) {

       List<Integer> myList =
Arrays.asList(10,15,8,49,25,98,98,32,15);
       Set<Integer> set = new HashSet();
       myList.stream()
               .filter(n -> !set.add(n))
               .forEach(System.out::println);
   }
}
```

Write a program to print the count of each character in a String? and remove white space

Input - `string data to count each character`
Output - {a=5, c=4, d=1, e=2, g=1, h=2, i=1, n=2, o=2, r=3, s=1, t=5, u=1}

```
public class A {
   public static void main(String[] args) {
       List<String> list = Arrays.asList("string data to
count each character");

       // Join the list into a single string
       String combinedString = list.stream().
               collect(Collectors.joining());
       Map<String, Long> collect =
Stream.of(combinedString.replace(" ", "").split(""))

.collect(Collectors.groupingBy(String::toLowerCase,
Collectors.counting()));
       System.out.println(collect);

   }
}
```

https://www.linkedin.com/in/kunalkr19