1. How do you find the missing number in a given integer array of 1 to 100?

```
import java.io.*;
import java.util.*;

class MissingNumber {
        public static void findMissing(int arr[], int N)
        {
                int i;
                int temp[] = new int[N + 1];
                for (i = 0; i <= N; i++) {
                        temp[i] = 0;
                }

                for (i = 0; i < N; i++) {
                        temp[arr[i] - 1] = 1;
                }

                int ans = 0;
                for (i = 0; i <= N; i++) {
                        if (temp[i] == 0)
                                ans = i + 1;
                }
                System.out.println(ans);
        }
                 public static void main(String[] args)
        {
                int arr[] = { 1,5,4,2,7,6 };
                int n = arr.length;
                findMissing(arr, n);
        }
}
```

2.How do you find the duplicate number on a given integer array?

```
public class DuplicateElement {
   public static void main(String[] args) {
      int [] arr = new int [] {1,2,3,6,4,2,5,3};
     System.out.println("Duplicate elements in given array: ");
     for(int i = 0; i < arr.length; i++) {
         for(int j = i + 1; j < arr.length; j++) {
            if(arr[i] == arr[j])
               System.out.println(arr[j]);
         }
      }
   }
}
```

3.How do you find the largest and smallest number in an unsorted integer array?

```
class Largest {
        static int[] findMinMax(int[] arr, int n)
        {
                int mini = arr[0];
                int maxi = arr[0];

                for (int i = 0; i < n; i++) {
                        if (arr < mini) {
                                mini = arr[i];
                        }
                        else if (arr[i] > maxi) {
                                maxi = arr[i];
                        }
                }
                int[] ans = new int[2];
                ans[0] = mini;
                ans[1] = maxi;
                return ans;
        }

        public static void main(String[] args)
        {
                int[] arr = { 1, 2, 3, 4, 5 };
                int N = arr.length;
                int[] ans = findMinMax(arr, N);
                System.out.print("Maximum is: " + ans[1]);
                System.out.print("\n"+ "Minimum is: " + ans[0]);
        }
}
```

4. How do you find all pairs of an integer array whose sum is equal to a given number?

```
public class Find {
        public static void main(String args[])
        {
                int[] arr = { 1, 5, 7, -1, 5 };
                int sum = 6;
                getPairsCount(arr, sum);
```

```
    }

    public static void getPairsCount(int[] arr, int sum)
    {

        int count = 0;
        for (int i = 0; i < arr.length; i++)
            for (int j = i + 1; j < arr.length; j++)
                if ((arr[i] + arr[j]) == sum)
                    count++;

        System.out.printf("Count of pairs is %d", count);
    }
}
```

5. How do you find duplicate numbers in an array if it contains multiple duplicates?

```
import java.util.ArrayList;
public class Duplicate{
    static void findDuplicates(
    int arr[], int len)
    {
        boolean ifPresent = false;
        ArrayList<Integer> al = new ArrayList<Integer>();

        for (int i = 0; i < len - 1; i++) {
            for (int j = i + 1; j < len; j++) {
                if (arr[i] == arr[j]) {
                    if (al.contains(arr[i])) {
                        break;
                    }

                    else {
                        al.add(arr[i]);
                        ifPresent = true;
                    }
                }
            }
        }
        if (ifPresent == true) {

            System.out.print(al + " ");
        }
        else {
            System.out.print("No duplicates present in arrays");
        }
    }
    public static void main(String[] args)
    {

        int arr[] = { 12, 11, 40, 12, 5, 6, 5, 12, 11 };
        int n = arr.length;

        findDuplicates(arr, n);
    }
}
```

6. How are duplicates removed from a given array in Java?

```
public class Main {

    public static int removeduplicates(int a[], int n)
    {
        if (n == 0 || n == 1) {
            return n;
        }
        int[] temp = new int[n];
        int j = 0;

        for (int i = 0; i < n - 1; i++) {
            if (a[i] != a[i + 1]) {
                temp[j++] = a[i];
            }
        }

        temp[j++] = a[n - 1];
        for (int i = 0; i < j; i++) {
            a[i] = temp[i];
        }

        return j;
    }
    public static void main(String[] args)
    {
        int a[] = { 1, 1, 2, 2, 2 };
        int n = a.length;

        n = removeduplicates(a, n);
        for (int i = 0; i < n; i++)
            System.out.print(a[i] + " ");
    }
}
```

7. How is an integer array sorted in place using the quicksort algorithm?

```
import java.io.*;
class QuickSort {
    static void swap(int[] arr, int i, int j)
    {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    static int partition(int[] arr, int low, int high)
    {

        int pivot = arr[high]{
        int i = (low - 1);
        for (int j = low; j <= high - 1; j++) {
            if (arr[j] < pivot){
                i++;
                swap(arr, i, j);
            }
        }
        swap(arr, i + 1, high);
        return (i + 1);
    }
    static void quickSort(int[] arr, int low, int high)
    {
        if (low < high) {
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }
    static void printArray(int[] arr, int size)
    {
        for (int i = 0; i < size; i++)
            System.out.print(arr[i] + " ");

        System.out.println();
    }

    public static void main(String[] args)
    {
        int[] arr = { 10, 7, 8, 9, 1, 5 };
        int n = arr.length;

        quickSort(arr, 0, n - 1);
        System.out.println("Sorted array: ");
        printArray(arr, n);
    }
}
```

8. How do you remove duplicates from an array in place?

```
class Main
{
.
    static int removeDuplicates(int arr[], int n)
    {
        if (n == 0 || n == 1)
            return n;
        int j = 0;
        for (int i = 0; i < n-1; i++)
```

```
            if (arr[i] != arr[i+1])
                arr[j++] = arr[i];

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

        return j;
    }

    public static void main (String[] args)
    {
        int arr[] = {1,3,3,2,2,4,4,5,5};
        int n = arr.length;

        n = removeDuplicates(arr, n);

        // Print updated array
        for (int i=0; i<n; i++)
        System.out.print(arr[i]+" ");
    }
}
```

9. How do you reverse an array in place in Java?

```
    public class ReverseArray {
    static void reverse(int a[], int n)
    {
        int[] b = new int[n];
        int j = n;
        for (int i = 0; i < n; i++) {
            b[j - 1] = a[i];
            j = j - 1;
        }

        System.out.println("Reversed array is: \n");
        for (int k = 0; k < n; k++) {
            System.out.println(b[k]);
        }
    }
```

```
        public static void main(String[] args)
        {
            int [] arr = {10, 20, 30, 40, 50};
            reverse(arr, arr.length);
        }
    }
```

10. How are duplicates removed from an array without using any library?

```
       class Main {
        static int removeDuplicates(int arr[], int n)
        {

            if (n == 0 || n == 1)
                return n;
             int[] temp = new int[n];
            int j = 0;
            for (int i = 0; i < n - 1; i++)
            temp[j++]  = arr[n  - 1];
            for (int i = 0; i < j; i++)
                arr[i] = temp[i];

            return j;
        }

        public static void main(String[] args)
        {
            int arr[] = { 1, 2, 2, 3, 4, 4, 4, 5, 5 };
            int n = arr.length;

            n = removeDuplicates(arr, n);
            for (int i = 0; i < n; i++)
                System.out.print(arr[i] + " ");
        }
    }
```

11. How do you print duplicate characters from a string?

```
public class GFG {
    static final int NO_OF_CHARS = 256;
```

```
        static void fillCharCounts(String str,int[] count)
        {
                for (int i = 0; i < str.length(); i++)
                        count[str.charAt(i)]++;
        }
        static void printDups(String str)
        {
                int count[] = new int[NO_OF_CHARS];
                fillCharCounts(str, count);

                for (int i = 0; i < NO_OF_CHARS; i++)
                        if (count[i] > 1)


                                System.out.println((char)(i) +", count = " + count[i]);
        }
        public static void main(String[] args)
        {
                String str = "test string";
                printDups(str);
        }
}
```

12. How do you check if two strings are anagrams of each other?

```
import java.io.*;
import java.util.*;

class Anagram {

        static int NO_OF_CHARS = 256;
        static boolean areAnagram(char str1[], char str2[])
        {
                int count1[] = new int[NO_OF_CHARS];
                Arrays.fill(count1, 0);
                int count2[] = new int[NO_OF_CHARS];
                Arrays.fill(count2, 0);
                int I;
                for (i = 0; i < str1.length && i < str2.length;
                        i++) {
                        count1[str1[i]]++;
```

```
                    count2[str2[i]]++;
            }
            if (str1.length != str2.length)
                    return false;
            for (i = 0; i < NO_OF_CHARS; i++)
                    if (count1[i] != count2[i])
                            return false;

            return true;
    }
    public static void main(String args[])
    {
            char str1[] = ("gram").toCharArray();
            char str2[] = ("arm").toCharArray();
            if (areAnagram(str1, str2))
                    System.out.println("The two strings are"+ " anagram of each other");
            else
                    System.out.println("The two strings are”+ " anagram of each other");
    }
}
```

13. How do you print the first non-repeated character from a string

```
 public class FirstNonRepeatedCharFirst {
  public static void main(String args[]) {
   String inputStr ="teeter";
   for(char i :inputStr.toCharArray()){
     if ( inputStr.indexOf(i) == inputStr.lastIndexOf(i)) {
       System.out.println("First non-repeating character is: "+i);
       break;
     }
     }
  }
}
```

14. How can a given string be reversed using recursion?

```
    class StringReverse

    {

            void reverse(String str)

            {
```

```
        if ((str==null)||(str.length() <= 1))

        System.out.println(str);

        else

        {

            System.out.print(str.charAt(str.length()-1));

            reverse(str.substring(0,str.length()-1));

        }

    }

    public static void main(String[] args)

    {

        String str = "Mounica";

        StringReverse obj = new StringReverse();

        obj.reverse(str);

    }

}
```

15. How do you check if a string contains only digits?

```
Class Digits {

    public static boolean

    onlyDigits(String str, int n)

    {

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

            if (str.charAt(i) < '0'

                || str.charAt(i) > '9') {

                return false;

            }

        }

        return true;

    }
```

```
      Public static void main(String args[])
      String str=”1a2345”;
      int length=str.length();
      System.out.println(onlyDigits(str,len));
     }
  }
```

16. How are duplicate characters found in a string?

```
import java.util.*;
 class Duplicate{
public static void
countDuplicateCharacters(String str)
{
        Map<Character, Integer> map= new HashMap<Character, Integer>();
        char[] charArray = str.toCharArray();
        for (char c : charArray) {
          if (map.containsKey(c)) {
                        map.put(c, map.get(c) + 1);
                }
                else
                        map.put(c, 1);
                }
        }
        for (Map.Entry<Character, Integer> entry :
                map.entrySet()) {
                  if (entry.getValue() > 1) {
                        System.out.println(entry.getKey()+ " : + entry.getValue());
                }
        }
```

```
    }
    public static void
    main(String args[])
    {
        String str = "Sivapriya";
        countDuplicateCharacters(str);
    }
}
```

17. How do you count the number of vowels and consonants in a given string?

```
    import java.io.*;
   public class Count  {
        static void countCharacterType(String str)
        {
            int vowels = 0, consonant = 0, specialChar = 0,
                digit = 0;
                for (int i = 0; i < str.length(); i++) {
                char ch = str.charAt(i);
                  if ( (ch >= 'a' && ch <= 'z') ||
                    (ch >= 'A' && ch <= 'Z') )
                    ch = Character.toLowerCase(ch);

                    if (ch == 'a' || ch == 'e' || ch == 'i' ||
                        ch == 'o' || ch == 'u')
                        vowels++;
                    else
                        consonant++;
                }
                else if (ch >= '0' && ch <= '9')
                digit++;
            else
                specialChar++;
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonant: " + consonant);
        System.out.println("Digit: " + digit);
        System.out.println("Special Character: " + specialChar);
    }
    static public void main (String[] args)
    {
        String str = "sivapriya1203 ";

        countCharacterType(str);
    }
}
```

18. How do you count the occurrence of a given character in a string?

```
class Occurence
{
    public static int count(String s, char c)
    {
        int res = 0;

        for (int i=0; i<s.length(); i++)
        {
            if (s.charAt(i) == c)
            res++;
```

```
                    }
                    return res;
          }
          public static void main(String args[])
          {
                    String str= "Sivapriya";
                    char c = 'a';
                    System.out.println(count(str, c));
          }
}
```

19. How do you reverse words in a given sentence without using any library method?

```
import java.util.*;
 class Permutation{
static void reverse(char str[],int start,int end)
 {
char temp;

 while (start <= end)
 {
          temp = str[start];
          str[start] = str[end];
          str[end] = temp;
          start++;
          end--;
 }
 }
 static char[] reverseWords(char []s)
 {
  int start = 0;

  for (int end = 0; end < s.length; end++)

  {

  if (s[end] == ' ')

            {

            reverse(s, start, end);

            start = end + 1;

            }

  }

reverse(s, start, s.length - 1);

reverse(s, 0, s.length - 1);

return s;

  }

public static void main(String[] args)

  {

  String s = "i like this program very much ";

  char []p = reverseWords(s.toCharArray());

  System.out.print(p);

  }

  }
```