51. How can you reverse the words in a target sentence without the help of library methods?

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    string str = "you shall not pass";
    cout << "Original string: " << str << endl;

    // Use the built-in reverse to get char-by-char reversal.
    reverse(str.begin(), str.end());

    string buffer = "";
    string ans = "";

    // This for loop then reverses each individual
    // word in the string.
    for (int i=0; i < str.length(); i++)
    {
      if (str[i] != ' ')
      {
        buffer += str[i];
      }
      else
      {
        reverse(buffer.begin(), buffer.end());
        ans += buffer + " ";
        buffer = "";
      }
    }
    // Reversing the last word in the string outside the loop:
    reverse(buffer.begin(), buffer.end());
    ans += buffer;
    cout << "Reversed string: " << ans << endl;
}
```

52. How can you replace or remove characters from strings?

```
s = "flexible!"
s2 = string.replace("b","p")

print(s2)
// Output - "flexiple!"

//Code to remove a character
s3 =string.replace("!","")

print(s3)
```

53. How can you append texts to files in programming languages such as Java?

```
import java.io.*;

class GeeksforGeeks {
    public static void appendStrToFile(String fileName,
    String str)
    {
        try {
            BufferedWriter out = new BufferedWriter(
                new FileWriter(fileName, true));

            out.write(str);
            out.close();
        }

        catch (IOException e) {

            System.out.println("exception occurred" + e);
        }
    }
```

54. How can you find the largest or smallest number in an array of integers?

```
import java.io.*;
public class GFG {
static String[] findLargestAndSmallest(String[] numbers)
{

    int N = numbers.length;
    int maxLen = 0, minLen = Integer.MAX_VALUE;
    String[] res = { "", "" };

    for (int i = 0; i < N; i++) {
    int numLen = numbers[i].length();
    if (minLen > numLen) {
        minLen = numLen;
        res[0] = numbers[i];
    }
    else if (minLen == numLen)
        res[0] = ((res[0].length()
                > numbers[i].length())
                ? res[0]
                : numbers[i]);
    if (maxLen < numLen) {
        maxLen = numLen;
        res[1] = numbers[i];
    }

    else if (maxLen == numLen)
        res[1]
```

```
                = (res[1].length() > numbers[i].length()
                ? res[1]
                : numbers[i]);
        }

        return res;
}

public static void main(String[] args)
{
        String[] numbers
        = { "5677856", "657856745356587687698768",
                "67564", "45675645356576578564435647647" };
        // Calling the function
        String[] ans = findLargestAndSmallest(numbers);
        System.out.println("Smallest: " + ans[0]);
        System.out.print("Largest: " + ans[1]);
}
}
```

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

```
import java.util.*;
import java.util.Arrays;

class GFG {
        public static int getMissingNo(int[] nums, int n)
        {
                int sum = ((n + 1) * (n + 2)) / 2;
                for (int i = 0; i < n; i++)
                        sum -= nums[i];
                return sum;
        }

        public static void main(String[] args)
        {
                int[] arr = { 1, 2, 3, 5 };
                int N = arr.length;
                System.out.println(getMissingNo(arr, N));
        }
}
```

56. How to find the duplicate number on a given integer array?

```
class Leet442 {

        public static void main(String args[])
        {
                int numRay[] = { 0, 4, 3, 2, 7, 8, 2, 3, 1 };

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

```
            numRay[numRay[i] % numRay.length]
                = numRay[numRay[i] % numRay.length]
                + numRay.length;
        }
        System.out.println("The repeating elements are : ");
        for (int i = 0; i < numRay.length; i++) {
            if (numRay[i] >= numRay.length * 2) {
                System.out.println(i + " ");
            }
        }
    }
}
```

57. How to find the largest and smallest number in an unsorted integer array?

```
public class FindLargestSmallestNumber {

 public static void main(String[] args) {

  //numbers array
  int numbers[] = new int[]{55,32,45,98,82,11,9,39,50};

  //assign first element of an array to largest and smallest
  int smallest = numbers[0];
  int largetst = numbers[0];

  for (int i = 1; i < numbers.length; i++) {
   if (numbers[i] > largetst)
    largetst = numbers[i];
   else if (numbers[i] < smallest)
    smallest = numbers[i];
  }

  System.out.println("Largest Number is : " + largetst);
  System.out.println("Smallest Number is : " + smallest);
 }
}
```

58. How to find all pairs of integer arrays 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; // Initialize result

        // Consider all possible pairs and check their sums
        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);
    }
}
```

59. How to find duplicate numbers in an array if it contains multiple duplicates?

```
#include <stdio.h>

int main()
{
    //Initialize array
    int arr[] = {1, 2, 3, 4, 2, 7, 8, 8, 3};

    //Calculate length of array arr
    int length = sizeof(arr)/sizeof(arr[0]);

    printf("Duplicate elements in given array: \n");
    //Searches for duplicate element
    for(int i = 0; i < length; i++) {
        for(int j = i + 1; j < length; j++) {
            if(arr[i] == arr[j])
                printf("%d\n", arr[j]);
        }
    }
    return 0;
}
```

60. How to remove duplicates from a given array?

```
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++)

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

```
                temp[j++] = arr[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);

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

61. How to sort an integer array in place using the QuickSort algorithm?

```
#include <stdio.h>
int partition (int a[], int start, int end)
{
  int pivot = a[end]; // pivot element
  int i = (start - 1);

  for (int j = start; j <= end - 1; j++)
  {

    if (a[j] < pivot)
    {
      i++;
      int t = a[i];
      a[i] = a[j];
      a[j] = t;
    }
  }
  int t = a[i+1];
  a[i+1] = a[end];
  a[end] = t;
  return (i + 1);
}
 {
  if (start < end)
  {
    int p = partition(a, start, end); //p is the partitioning index
```

```
        quick(a, start, p - 1);
        quick(a, p + 1, end);
    }
}
  void printArr(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}
int main()
{
    int a[] = { 24, 9, 29, 14, 19, 27 };
    int n = sizeof(a) / sizeof(a[0]);
    printf("Before sorting array elements are - \n");
    printArr(a, n);
    quick(a, 0, n - 1);
    printf("\nAfter sorting array elements are - \n");
    printArr(a, n);

    return 0;
}
```

62. How to remove duplicates from an array in place?

```
public class RemoveDuplicateInArrayExample{
public static int removeDuplicateElements(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++){
            if (arr[i] != arr[i+1]){
                temp[j++] = arr[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[] = {10,20,20,30,30,40,50,50};
        int length = arr.length;
        length = removeDuplicateElements(arr, length);
        for (int i=0; i<length; i++)
           System.out.print(arr[i]+" ");
```

63. How to 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);
        }
}
```

64. How to find multiple missing numbers in a given integer array with duplicates?

```
import java.util.*;

class GFG{
        static void printMissingElements(int arr[],int N)
{
        int diff = arr[0] - 0;

        for(int i = 0; i < N; i++)
        {
                if (arr[i] - i != diff)
                {
                        while (diff < arr[i] - i)
                        {
                                System.out.print((i + diff) + " ");
                                diff++;
                        }
                }
        }
}
public static void main (String[] args)
{

        int arr[] = { 6, 7, 10, 11, 13 };
```

```
             int N = arr.length;
             printMissingElements(arr, N);
}
}
```

65. How to Print duplicate characters from String?

```
#include <stdio.h>
#include <string.h>

int main()
{
    char string[] = "Great responsibility";
    int count;

    printf("Duplicate characters in a given string: \n");
    //Counts each character present in the string
    for(int i = 0; i < strlen(string); i++) {
        count = 1;
        for(int j = i+1; j < strlen(string); j++) {
            if(string[i] == string[j] && string[i] != ' ') {
                count++;
                //Set string[j] to 0 to avoid printing visited character
                string[j] = '0';
            }
        }
        //A character is considered as duplicate if count is greater than 1
        if(count > 1 && string[i] != '0')
            printf("%c\n", string[i]);
    }

    return 0;
}
```

66. How to check if two Strings are anagrams of each other?

```
import java.util.Arrays;

public class AnagramString {
    static void isAnagram(String str1, String str2) {
        String s1 = str1.replaceAll("\\s", "");
        String s2 = str2.replaceAll("\\s", "");
        boolean status = true;
        if (s1.length() != s2.length()) {
            status = false;
        } else {
            char[] ArrayS1 = s1.toLowerCase().toCharArray();
            char[] ArrayS2 = s2.toLowerCase().toCharArray();
            Arrays.sort(ArrayS1);
```

```
      Arrays.sort(ArrayS2);
      status = Arrays.equals(ArrayS1, ArrayS2);
    }
    if (status) {
      System.out.println(s1 + " and " + s2 + " are anagrams");
    } else {
      System.out.println(s1 + " and " + s2 + " are not anagrams");
    }
  }

  public static void main(String[] args) {
    isAnagram("Keep", "Peek");
    isAnagram("Mother In Law", "Hitler Woman");
  }
}
```

67. How to print the first non-repeated character from 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;
    }
    }
  }
}
```

68. How to reverse a given string using recursion?

```
// Java program to reverse a string using recursion

class StringReverse
{
    /* Function to print reverse of the passed string */
    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));
        }
    }

    /* Driver program to test above function */
    public static void main(String[] args)
    {
        String str = "Geeks for Geeks";
        StringReverse obj = new StringReverse();
        obj.reverse(str);
    }
}
```

69. How to check if a string contains only digits?

```
// Java program for the above approach
// contains only digits
class GFG {

    // Function to check if a string
    // contains only digits
    public static boolean
    onlyDigits(String str, int n)
    {
        // Traverse the string from
        // start to end
        for (int i = 0; i < n; i++) {

            // Check if character is
            // not a digit between 0-9
            // then return false
            if (str.charAt(i) < '0'
                || str.charAt(i) > '9') {
                return false;
            }
        }
        // If we reach here, that means
        // all characters were digits.
        return true;
    }

    // Driver Code
    public static void main(String args[])
    {
        // Given string str
        String str = "1a234";
        int len = str.length();

        // Function Call
        System.out.println(onlyDigits(str, len));
    }
}
```

70. How to find duplicate characters in a String?

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

	/* Fills count array with
	frequency of characters */
	static void fillCharCounts(String str,
											int[] count)
	{
		for (int i = 0; i < str.length(); i++)
			count[str.charAt(i)]++;
	}

	/* Print duplicates present
	in the passed string */
	static void printDups(String str)
	{
		// Create an array of size
		// 256 and fill count of
		// every character in it
		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]);
	}

	// Driver Method
	public static void main(String[] args)
	{
		String str = "test string";
		printDups(str);
	}
}
```