Strings

1.) Java program to reverse a string

```
import java.util.Scanner;
 public class Test {
       public static void main(String[] args) {
          Scanner scanner = new Scanner(System.in);
          System.out.print("Enter a string: ");
          String input = scanner.nextLine();
          char ch;
          String nstr = "";
          for (int i = 0; i < input.length(); i++) {
                ch = input.charAt(i);
                nstr = ch + nstr;
            }
 System.out.println("Reversed String is : " + nstr);
```

2.) Java program to reverse each word of a given string

```
public static void main(String[] args) {
      reverseEachWordOfString("Java is good programming langauges");
}

static void reverseEachWordOfString(String inputString)
{
      String[] words = inputString.split(" ");

      String reverseString = "";
      for (int i = 0; i < words.length; i++) {
                String word = words[i];
                String nstr = "";
                char ch;
                for (int j = 0; j < word.length(); j++) {
                          ch = word.charAt(j);
                          nstr = ch + nstr;
      }
      reverseString = reverseString + nstr + " ";
}

      System.out.println(inputString);
      System.out.println(reverseString);
}
```

Input: Java is good programming langauges

Output: avaJ si doog gnimmargorp seguagnal

3.) Java program to find duplicate characters in a string

```
import java.util.HashMap;
import java.util.Set;

public class Main {

public static void main(String[] args) {
   duplicateCharacterCount("Learn Java Programming");
}

static void duplicateCharacterCount(String inputString) {

HashMap<Character, Integer> charCountMap = new HashMap<>();
char[] strArray = inputString.toCharArray();
for (char c : strArray) {
    if (charCountMap.containsKey(c)) {
        charCountMap.put(c, charCountMap.get(c) + 1);
    } else {
        charCountMap.put(c, 1);
    }
}

Set<Character> charsInString = charCountMap.keySet();
System.out.println("Duplicate Characters in : " + inputString);
for (Character ch : charsInString) {

    if (charCountMap.get(ch) > 1) {
        System.out.println(ch + " : " + charCountMap.get(ch));
    }
}
}
}
```

Duplicate Characters in : Learn Java Programming a : 4

g : 2 m : 2 n : 2 r : 3

4.) Java program to count Occurrences of Each Character in String

```
import java.util.HashMap; public class Main {


        public static void main(String[] args) {
              CharacterCount("Test Automation Java Automation");
        }

        static void CharacterCount(String inputString) {
           HashMap<String,Integer> charCountMap = new HashMap<>();
           for(String s : inputString.split(" "))
           {
                 if(charCountMap.containsKey(s))
                 { }
                 else charCountMap.put(s,charCountMap.get(s)+1);
                 { }


                       charCountMap.put(s,1);

           }
           System.out.println("Count of Characters in a given string : " +
    charCountMap);
                 }
           }
Count of Characters in a given string : {Java=1, Automation=2, Test=1}
```

5.) Java program to count the number of words in a string

```
 public class Main {
     public static void main(String[] args) {
   System.out.println("Enter the String");
   Scanner sc = new Scanner(System.in);
   String s = sc.nextLine();
   int count = 1;
   for (int i = 0; i < s.length() - 1; i++) {

         if ((s.charAt(i) == ' ') && (s.charAt(i + 1) != ' ')) {
               count++;
         }
   }
   System.out.println("Number of words in a string: " +count);              }
}
Enter the String: Welcome to Java World
Number of words in a string: 4
```

6.) Java program to find all permutations of a given string

```
import java.util.Scanner;

  public class Main {
      public static void main(String[] args) {
          String str = "abc";
          permute(str, "");
      }

      static void permute(String str, String prefix) {
            if (str.length() == 0) {
                  System.out.println(prefix);
            } else {
                  for (int i = 0; i < str.length(); i++) {
                        String rem = str.substring(0,i) + str.substring(i+1);
                        permute(rem,prefix + str.charAt(i));
                  }
            }
      }
}
```

abc

acb

bac

bca

cab

cba

7.) Java program to find if a string is Palindrome

```
import java.util.Scanner;

public class Main {
	public static void main(String[] args) {
		String str = "madam";
		System.out.println(isPalindrome(str));
	}

	static boolean isPalindrome(String str) {
		int start = 0;
		int end = str.length() - 1;

		while (start < end) {
			if (str.charAt(start) != str.charAt(end)) {
				return false;
			}
			start++;
			end--;
		}
		return true;
	}
}
```

8.) Java program to determine if Two Strings are Anagrams

```
public class Main {

	public static void main(String[] args) {
		String str1 = "listen";
		String str2 = "silent";
		System.out.println(areAnagrams(str1,str2));
	}

	static boolean areAnagrams(String str1, String str2) {
		if(str1.length() != str2.length())
		{ }
			return false;
		int[] charCount = new int[256];
		for( int i = 0; i < str1.length(); i++)
		{


			charCount[str1.charAt(i)]++;
			charCount[str2.charAt(i)]--;
		}
		for ( int count : charCount)
		{
			if ( count !=0 )
			{ }
				return false;

		}
		return true;
	}
}
```

9.) Java program to Count Vowels and Consonants in a given string

```
public class Main {
    public static void main(String[] args) {
            String str = "Hello World";
            VowelConsonantCount(str);
    }

    static void VowelConsonantCount(String str) {
        int vowels = 0, consonants = 0;
        str = str.toLowerCase();
        for (char c : str.toCharArray()) {
            if (c >= 'a' && c <= 'z') {
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
{
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }
        System.out.println("Vowels : " + vowels);
        System.out.println("Consonants : " + consonants);
    }
}
```

Vowels : 3

Consonants : 7

10.) Java program to print unqiue characters

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        System.out.println("Unique characters in \"" + input + "\":");
        printUniqueCharacters(input);

    }



    public static void printUniqueCharacters(String str) {
        // Assume ASCII characters (0-127), use boolean array to track
character occurrences
        boolean[] unique = new boolean[128];
        for (int i = 0; i < str.length(); i++) {

            char ch = str.charAt(i);
            if (!unique[ch]) {
                unique[ch] = true;
                System.out.print(ch + " ");
            }
        }

    }
}
```

Enter a string: **Java Automation**

Unique characters in "Java Automation":

J a v A u t o m i n

11.) Java program to print even indexed characters

```
import java.util.Scanner;

public class Main {
public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

  System.out.println("Even indexed characters in \"" + input + "\":");
printEvenIndexedCharacters(input);

}


public static void printEvenIndexedCharacters(String str) {
    for (int i = 0; i < str.length(); i++) {
        if (i % 2 == 0) {
            System.out.print(str.charAt(i));
        }
    }

}
}
```

Enter a string: Automation

Even indexed characters in "Automation":

Atmto

12.) Java program to remove space from a given string

```
import java.util.Scanner;

public class Main {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter a string with spaces: ");
		String input = scanner.nextLine();

		String stringWithoutSpaces = removeSpaces(input);
		System.out.println("String without spaces: " +
stringWithoutSpaces);
	}
	public static String removeSpaces(String str) {
	StringBuilder result = new StringBuilder();
	for (int i = 0; i < str.length(); i++) {
			if (str.charAt(i) != ' ') {
				result.append(str.charAt(i));
			}
		}
		return result.toString();
	}
}
```

Enter a string with spaces: Welcome to Java World
String without spaces: **WelcometoJavaWorld**

13.) Java program to print each letter twice from a given string

```
import java.util.Scanner;

public class Main {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter a string: ");
		String input = scanner.nextLine();

		String doubledString = doubleCharacters(input);
		System.out.println("Doubled characters: " + doubledString);
	}

	public static String doubleCharacters(String str) {

		StringBuilder doubled = new StringBuilder();
		for (int i = 0; i < str.length(); i++) {
			char ch = str.charAt(i);
			doubled.append(ch).append(ch); // Append each character
twice
		}
		return doubled.toString();
	}
}
```

```
Enter a string: hello
Doubled characters: hheelllloo
```

14.) Java program to swap two string without using 3rd variable

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter first string: ");
            String str1 = scanner.nextLine();
            System.out.print("Enter second string: ");
            String str2 = scanner.nextLine();

            System.out.println("Before swapping: str1 = " + str1 + ",
str2 = " + str2);

            // Swapping without using a third variable
            str1 = str1 + str2; // Concatenate str1 and str2 and
store in str1
            str2 = str1.substring(0, str1.length() - str2.length());
// Extract the initial part (original str1) from the concatenated
string
            str1 = str1.substring(str2.length()); // Extract the
remaining part (original str2) from the concatenated string

            System.out.println("After swapping: str1 = " + str1 + ",
str2 = " + str2);
        }
}
```

Enter first string: Hello

Enter second string: World

Before swapping: str1 = **Hello**, str2 = **World**

After swapping: str1 = **World**, str2 = **Hello**

15.) Java program to gives Output: a2b2c3d2 for the Input String Str = “aabbcccdd”

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String output = getCharacterCount(input);
        System.out.println("Output: " + output);
    }


    public static String getCharacterCount(String str) {
        StringBuilder result = new StringBuilder();
        int count = 1;

        for (int i = 0; i < str.length(); i++) {
            // If the next character is the same, increase the count
            if (i + 1 < str.length() && str.charAt(i) == str.charAt(i
+ 1)) {
                count++;
            } else {
                // Append the character and its count to the result
                result.append(str.charAt(i)).append(count);
                count = 1; // Reset the count
            }
        }

        return result.toString();
    }
}
```

Enter a string: aabbcccdd

Output: a2b2c3d2

16.) Java program to gives two Output:

“abcde”, “ABCDE” for the Input String Str = “aBACbcEDed”

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
                        System.out.println("Original String is: "+ input);
                        separateCharacters(input);
        }

     public static void separateCharacters(String input)
        {
         StringBuilder lowerCase = new StringBuilder();
         StringBuilder upperCase = new StringBuilder();
         for(char ch : input.toCharArray())
         {
               if(Character.isLowerCase(ch))
               { }
               else lowerCase.append(ch);
               { }


                    upperCase.append(ch);

         }
         System.out.println("Output in lowercase: "+lowerCase);
         System.out.println("Output in uppercase "+upperCase);
    }
```

Enter a string: **aBACbcEDed**

Output in lowercase: abced

Output in uppercase: ABCED

17.) Java program to gives two Output: “Subburaj”, “123” for the Input String Str = “Subbu123raj”

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
                    System.out.println("Original String is: "+ input);
                    separateAplhaAndNumeric(input);
    }

    public static void separateAlphaAndNumeric(String input)
    {
        StringBuilder alphaPart = new StringBuilder();
        StringBuilder numericPart = new StringBuilder();
        for(char ch : input.toCharArray())
        {
            if(Character.isLetter(ch))
            { }
            else if (Character.isDigit(ch))
            { }

                alphaPart.append(ch);
                numericPart.append(ch);

        }

        System.out.println("Output in Alpha: "+alphaPart.toString());
        System.out.println("Output in Numeric:
"+numericPart.toString());
    }
```

Enter a string: **Subbu123raj**

Output in lowercase: Subburaj

Output in uppercase: 123

18.) Java program to gives Output: “32412120000” for the Input String Str = “32400121200”

```
public class Main {
    public static void main(String[] args) {
        String input = "32400121200";
        String output = rearrangeDigits(input);
        System.out.println("Output: " + output);
    }

    public static String rearrangeDigits(String input) {
        // Split the input into parts: digits and non-digits
        StringBuilder digits = new StringBuilder();
        StringBuilder nonDigits = new StringBuilder();
        for (char c : input.toCharArray()) {
            if (Character.isDigit(c)) {
                digits.append(c);
            } else {
                nonDigits.append(c);
            }
        }

        // Concatenate non-digits followed by digits
        return digits.toString() + nonDigits.toString();
    }
}
```

Output: 32412120000

19.) Java program to gives Output: “00003241212” for the Input String Str = “32400121200”

```
public class Main {
    public static void main(String[] args) {
        String input = "32400121200";
        String formattedOutput = String.format("%011d",
Long.parseLong(input));
        System.out.println("Formatted output: " + formattedOutput);
    }
}
```

Formatted output: 00003241212

20.) Java program to find the longest without repeating characters

```
import java.util.HashSet;

public class Main {
	public static void main(String[] args) {
		String s1 = "abcabcbb"; // Expected: "abc", length 3
		String s2 = "bbbbb";	// Expected: "b", length 1
		String s3 = "pwwkew";	// Expected: "wke", length 3
		String s4 = "";		// Expected: "", length 0

		System.out.println("Longest substring without repeating
characters in s1: " + lengthOfLongestSubstring(s1)); // Output: 3
		System.out.println("Longest substring without repeating
characters in s2: " + lengthOfLongestSubstring(s2)); // Output: 1
		System.out.println("Longest substring without repeating
characters in s3: " + lengthOfLongestSubstring(s3)); // Output: 3
		System.out.println("Longest substring without repeating
characters in s4: " + lengthOfLongestSubstring(s4)); // Output: 0
	}

	public static int lengthOfLongestSubstring(String s) {
		HashSet<Character> set = new HashSet<>();
		int maxLength = 0;
		int start = 0;
		int end = 0;

		while (end < s.length()) {
			char currentChar = s.charAt(end);
			if (!set.contains(currentChar)) {
				set.add(currentChar);
				maxLength = Math.max(maxLength, end - start + 1);
				end++;
			} else {
				set.remove(s.charAt(start));
				start++;
			}
		}

		return maxLength;
	}
}
```