1. WAP(Write a Program) to remove Duplicates from a String.(Take any String example with duplicates Character.

Ans

```
import java.util.HashSet;


public class Demo {
   public static void main(String[] args) {
      String str1 = "HHeelloo";
      char []chararray =str1.toCharArray();


      StringBuilder result = new StringBuilder();
      HashSet<Character> hs = new HashSet<Character>();


      for (char c : chararray) {
         if(hs.add(c)){
            result.append(c);
         }
      }
      System.out.println(result);
   }
}
```

2. WAP to print Duplicates characters from the String.

Ans

```
import java.util.HashSet;


public class Demo {
   public static void main(String[] args) {
      String str1 = "HHeellloo";
      str1= str1.replace(" ","");
      char []chararray =str1.toCharArray();

      StringBuilder result = new StringBuilder();
      HashSet<Character> hs = new HashSet<Character>();

      for (char c : chararray) {
         if(hs.add(c)){
         }
         else{

            result.append(c);
         }
      }
      System.out.println(result);
   }
}
```

3. WAP to check if “2552” is palindrome or not?

Ans

```
class Demo{
public static void main(String[] args) {
String str1 ="2552";
String str2 = "";

for (int i = str1.length()-1;i>=0;i--) {
str2 = str2+str1.charAt(i);
}
if(str1.equals(str2))
{
   System.out.println("It is Palindrome");
}
else
{
   System.out.println("It is not a Palindrome");
}
}
}
```

4. WAP to count the number of consonants, vowels, special characters in a String

Ans

```
import java.util.Scanner;

public class Demo {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();
        str = str.toLowerCase();

        int vowels = 0, consonants = 0, digits = 0, specialChars = 0;

        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (Character.isLetter(ch)) {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            } else if (Character.isDigit(ch)) {
                digits++;
            } else {
                specialChars++;
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
        System.out.println("Digits: " + digits);
        System.out.println("Special Characters: " + specialChars);
    }
}
```

5. WAP to implement Anagram Checking least inbuilt methods being used?

Ans

```
import java.util.*;
public class stringAnagram {
   public static void main(String[] args) {
      String str1 = "Master school";
      String str2 = "The classroom ";

      str1= str1.replace(" ", "");
      str2= str2.replace(" ", "");

      str1=str1.toLowerCase();
      str2=str2.toLowerCase();

      char ar1[]=str1.toCharArray();
      char ar2[]=str2.toCharArray();

      Arrays.sort(ar1);
      Arrays.sort(ar2);

      if (Arrays.equals(ar1, ar2)) {
         System.out.println("It is an Anagram");
      } else {
         System.out.println("It is not an Anagram");
      }
   }
}
```

6. WAP to implement Pangram Checking with least inbuilt methods being used?

Ans

```
import java.util.*;


public class stringPangram {
   public static void main(String[] args) {
      boolean flag = false;
      String str = "The quick brown fox jumps over lazy dog";
      str = str.replace(" ", "");
      str = str.toUpperCase();
      char ch[] = str.toCharArray();
      Arrays.sort(ch);
      int ar1[] = new int[26];
      for (int i = 0; i < ch.length; i++) {
         ar1[ch[i] - 65]++;
      }
      for (int i = 0; i < ar1.length; i++) {
         if (ar1[i] == 0) {
            System.out.println("It is not a Pangram");
            flag = true;
         }
      }
      if (flag == false) {
         System.out.println("It is a Pangram");
      }
   }
}
```

7. WAP to find if String contains all unique characters?

Ans

```
import java.util.Scanner;
import java.util.*;

public class Demo {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a string: ");
      String str = scanner.nextLine();
      str = str.toUpperCase();
      HashSet<Character> set = new HashSet<Character>();

      for (int i= 0;i<str.length();i++){
         set.add(str.charAt(i));
      }

      boolean flag=set.size()==str.length();

      if (flag==true) {
         System.out.println("Yes , String contains all unique characters");
      } else {
         System.out.println("No , String do not contain all unique characters");
      }
   }
}
```

8. WAP to find the maximum occurring character in a String ?

Ans

```
import java.text.StringCharacterIterator;
import java.util.*;
public class Demo {
   public static void main(String[] args) {
      String str = "This is my String";
      char[] string = str.toCharArray();
      int[] Freq = new int[str.length()];

      for (int i = 0; i < string.length; i++) {
         Freq[i] = 1;
         for (int j = i + 1; j < Freq.length; j++) {
            if (string[i] == string[j] && string[i] != 0 && string[i] != ' ') {
               Freq[i]++;
               string[j] = '0';
            }
         }
      }
      int min, max;
      char minChar,maxChar;
      minChar=maxChar=str.charAt(0);
      min = max = Freq[0];
      for (int i = 0; i < Freq.length; i++) {
         if (min > Freq[i]) {
            min=Freq[i];
            minChar=string[i];
```

```
            } else if(max<Freq[i]) {
                max= Freq[i];
                 maxChar=string[i];
            }
        }
        System.out.println("Minimun occurence of character '"+minChar +"'' is "+min+" times");
        System.out.println("Maximun occurence of character '"+maxChar +"'' is "+max+" times");

    }
}
```