1. WAP to remove Duplicates from a String.

```
public class Dupli {

    public static void main(String[] args) {
        String str = "coconut";

        StringBuilder sb1 =new StringBuilder();
        for(int i=0;i<str.length();i++)
        {
            char ch= str.charAt(i);
            int indx =str.indexOf(ch , i+1);

            if(indx==-1)
            {
                sb1.append(ch);
            }

        }
        System.out.println("After removing 
duplicates:"+sb1);
}

}
```

2. WAP to print Duplicates characters from the String.

```
public class DupliChar {

    public static void main(String[] args) {

        String str = "Rabbit";
        int length=str.length();
        char ch [] = str.toCharArray();
        for (int i = 0; i <length ; i++)
        {

         for (int j = i+1; j <length; j++)
         {
          if (ch[i] == ch[j])
          {

            System.out.print(" Duplicate
characters are: " + ch[j] );
        }
    }
}
```

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

```
public class PalindromeAssign {

    public static void main(String[] args) {
        int num = 2552, rev = 0, rem;
        int originalNum = num;
        while (num != 0) {
          rem = num % 10;
          rev = rev * 10 + rem;
          num /= 10;
        }
        if (originalNum == rev) {
          System.out.println(originalNum + " is
Palindrome.");
        }
        else {
          System.out.println(originalNum + " is
not Palindrome.");
        }
}
}
}
```

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

```
public class Vowels {

    public static void main(String[] args) {
    String str = "My bike number is 4855";
    int vowels = 0, consonant = 0, digit = 0,
space = 0;
    str = str.toLowerCase();
    for (int i = 0; i < str.length(); ++i)
    {
    char ch = str.charAt(i);

    if (ch == 'a' || ch == 'e' || ch == 'i' ||
            ch == 'o' || ch == 'u')
        {
                    ++vowels;
        }
    else if ((ch >= 'a' && ch <= 'z'))
    {
    ++consonant;
    }

    // 0 to 9
    else if (ch >= '0' && ch <= '9')
    {
```

```
    ++digit;
    }

    else if (ch == ' ')
    {
    ++space;
    }
    }
    System.out.println("Vowels: " + vowels);
    System.out.println("Consonants: " +
consonant);
    System.out.println("Digits: " + digit);
    System.out.println("White spaces: " +
space);
}
```

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

```
import java.util.Arrays;

class AnagramAss {
  public static void main(String[] args) {
    String str1 = "Listen";
    String str2 = "Silent";

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

    if(str1.length() == str2.length()) {
      char[] charArray1 = str1.toCharArray();
      char[] charArray2 = str2.toCharArray();
      Arrays.sort(charArray1);
      Arrays.sort(charArray2);
      boolean result = Arrays.equals(charArray1,
charArray2);

      if(result)
      {
        System.out.println(str1 + " and " +
str2 + " are anagram.");
      }
      else
      {
        System.out.println(str1 + " and " +
str2 + " are not anagram.");
      }
    }

  }
}
```

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

```
import java.util.Scanner;
public class PanagramAss {
public static void main(String args[]){

    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Your String:");
    String str=sc.nextLine();
    str=str.replaceAll("","").toLowerCase();
    String s="";

        for(char i='a';i<='z';i++){
            if(str.indexOf(i)!=-1){

                s=s+i;
            }
        }
                if(s.length()==26){
            System.out.println("Pangram");
        }
        else{
           System.out.println(" Not  Pangram");
        }
    }
}
```

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

```
import java.util.Scanner;

public class Uniquechar {

   public static void main(String args[]) {

      Scanner sc = new Scanner(System.in);
      System.out.println("Enter String: ");
      String str = sc.next();
      System.out.println("Enter the required
character: ");
      char ch = sc.next().toCharArray()[0];

      int i = str.indexOf(ch);
      if(i!=-1)
      {
         System.out.println("Sting contains
uniquechar");
      }
      else
      {
         System.out.println("String doesn’t
contain uniquechar");
      }

    }

}
```

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

```
import java.util.Scanner;

public class Maxchar {

public static void main(String[] args) {

        String maxStr;
        char maxChar = ' ';
        int i, max = -1;
        int[] charFreq = new int[256];

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter String");
        maxStr = sc.nextLine();

        for(i = 0; i < maxStr.length(); i++)
        {
            charFreq[maxStr.charAt(i)]++;
        }
        for(i = 0; i < maxStr.length(); i++)
        {
            char ch = maxStr.charAt(i);
            if(max < charFreq[ch]) {
                max = charFreq[ch];
                maxChar = ch;
            }
        }
        System.out.println("The Maximum
Character is = " +  maxChar);

    }

}
```