1. WAP(Write a program) to remove Duplicates from a string (take any String example

With duplicates character).

Ans:

```
public class lecture_16 {
    public static void removeDuplicate(char[]c , int n){
     String res="";
    int i,j;
     for( i=0; i<n;i++){
         for( j=0; j<i;j++){
             if(c[i]==c[j]){
                 break;
             }
         }
         if(j==i){
             res+=c[i];
         }
     }
         System.out.print(res);
    }
    public static void main(String[] args) {
         String str= "rajankumar";
         char[]c= str.toCharArray();
         int n = c.length;
         removeDuplicate(c,n);
    }
}
```

2. WAP to print Duplicates characters from the String

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {
        String str= "rajankumar";
        char[]c= str.toCharArray();
        int n = c.length;

            for( int i=0; i<n;i++) {
                for (int j = 0; j < i; j++) {
                    if (c[i] == c[j]) {
                        System.out.print(c[i] + " ");
                        break;
                    }
                }
            }


    }
}
```

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

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {
            String str1= "2552";
        String str2 = "";
        for(int i= str1.length()-1; i>=0; i--){
            str2 += str1.charAt(i);
        }
        if(str1.equals(str2)){
            System.out.println("String is palindrome :");
        }else {
            System.out.println("NOT palindrome ");
        }
        System.out.println("Before Reversing: " + str1);
        System.out.println("After Reversing: " +str2);

    }
}
```

o/p:

```
String is palindrome :
Before Reversing: 2552
After Reversing: 2552
```

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

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {

        String str="I am rajankuamr from bihar";
        int v=0, c=0, s=0;  //v=vowel, c=constant, s=specialchracter
        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'){
                v++;
            }
            else if('a'<=ch && ch<='x'){
                c++;
            }else {
                s++;
            }
        }
        System.out.println(v);
        System.out.println(c);
        System.out.println(s);

    }
}
```

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

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {

     String str1 = "School Master";
        String str2 = "The Classroom";
        str1 = str1.replace(" ", "");
        str2 = str2.replace(" ", "");

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

        char arr1[] = str1.toCharArray();
        char arr2[] = str2.toCharArray();

        Arrays.sort(arr1);
        Arrays.sort(arr2);

        if(Arrays.equals(arr1,arr2)){
            System.out.println("IT IS AN ANAGRAM :");
        }else {
            System.out.println("NOT ANAGRAM :");
        }

    }
}
```

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

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {

    boolean flag= false;
        String str = "THE QUICK ROWN FOX JUMPS OVER LAZY DOG";
        str = str.replace(" ","");
         // to convert upper or lover case
            char []ch=str.toCharArray();

            int []ar= new int[26];
            for(int i=0; i<ch.length;i++){
                ar[ch[i]-65]++;
            }
            for(int i=0;i<ar.length;i++){
                if(ar[i]==0){
                    System.out.println("not pangram");
                    flag= true;
                }
            }
            if(flag==false){
                System.out.println("its pangram");
            }

    }
}
```

7. WAP to find if the string contains all unique characters.

```
import java.util.Arrays;

public class lecture_16 {

    public static boolean is_Unique_str(String str) {
        str = str.toLowerCase();
        char []ch= str.toCharArray();
        Arrays.sort(ch);
        for(int i=0;i<ch.length; ++i){
            if(ch[i]== ch[i-1]){
                return false;
            }
        }
         return true;
    }
    public static void main(String[] args) {
        String str="I am rajankuamr from bihar";
        System.out.println(is_Unique_str(str));

    }
}
```

8. WAP to find the maximum occurring characters in a string.

Ans:

```
public class lecture_16 {

    public static void main(String[] args) {

        String str ="i am rajankuamr";
        char []ch = str.toCharArray();
        Arrays.sort(ch);
        String s = new String(ch);

        int max = 0;
        int count = 1;
        char ans = '-';
        for (int i = 1; i <= s.length(); i++) {
            if ((i == s.length()) || (s.charAt(i) != s.charAt(i - 1))) {
                if (max < count) {
                    max = count;
                    ans = s.charAt(i-1);
                }count = 1;
            } else {
                count++;
            }
        }
        System.out.println("Maximum occurring character is "+ans);


    }
}
```