Q1. Write a Program to remove Duplicates from a String?

Ans.
```java
import java.lang.*;


class RemoveDuplicate {
    public static void main(String[] args) {
        String str="Hello, World!";
        char[] ch = str.toCharArray();
        int m=str.length();

        for(int i=0;i<m-1;i++){
            for(int j=i+1;j<m;j++){
                if(ch[i]==ch[j]){
                    for(int k=j;k<m-1;k++){
                        ch[k]=ch[k+1];
                    }


                }
            }
        }
        for(int n=0;n<m ;n++){
            System.out.print(""+ch[n]);
        }



    }
}
```

Q2. WAP to print Suplicates charecters from the string?

Ans
```java
class RemoveDuplicate {

    public static void main(String[] args) {
        String str="Hello, World!";
        char[] ch = str.toCharArray();
        int m=str.length();

        for(int i=0;i<m-1;i++){
            for(int j=i+1;j<m;j++){
                if(ch[i]==ch[j]){
                    System.out.print(ch[j]);
                    }



                }
            }
        }
        // for(int n=0;n<m ;n++){
        //     System.out.print(""+ch[n]);
        // }



    }
```

Q3. WAP to check if ”2552” is plaindrom aur not?

Ans.
```java
public class palindrome {

    public static void main(String[] args) {
        String str="2552";
        String str1="";
        for (int i = str.length()-1; i >=0; i--) {
```

```
            str1+=str.charAt(i);



        }
        if (str.equals(str1)) {
            System.out.println("it's Palindrome");

           }
           else
           {
               System.out.println("It's not the palindrome");
           }
    }
}
```

Q4. WAP to count constent,vowels and Special characters in a string?

Ans.
```
public class countCVS {

    public static void main(String[] args) {
        String str = "HEllo this is java course!!!!!!!!@@@##";
        str=str.toLowerCase();
        int num = 0;
        int sp = 0;
        int vowels = 0;
        int constent = 0;
        int specialChars = 0;
        char ch;
        for ( int i = 0; i < str.length(); i++) {
            ch=str.charAt(i);
            if (ch>='a' && ch<='z') {
                if (ch=='a' || ch=='e' || ch=='i' || ch=='o' ||ch=='u')
                vowels++;
                else
                constent++;
            }
            else if (ch>='0' &&ch<='9'){
            num++;
            }
            else if (ch==' '){
            sp++;
            }
            else
            {
                specialChars++;
            }

        }
       System.out.println("total number of vowels is: "+vowels);
       System.out.println("total number of constents is: "+constent);
       System.out.println("total number of space is: "+sp);
       System.out.println("total number of numbers is: "+num);
       System.out.println("total number of Spcial characters is: "+specialChars);
    }

}
```

Q5. Write a program implement Anagram Cheching with least inbuilt methods being used?

Ans
```
import java.util.Arrays;


public class annagramWithInBuild {
    public static void main(String[] args) {
        String str1="listen  ";
        String str2="SIlent";
```

```
        str1=str1.toLowerCase();
        str2=str2.toLowerCase();
        str1=str1.replace(" ", "");
        str2=str2.replace(" ", "");
    char arr[]=str1.toCharArray();
    char arr1[]=str2.toCharArray();
    Arrays.sort(arr);
    Arrays.sort(arr1);
    if (Arrays.equals(arr, arr1)) {
        System.out.println("It's a annagram String");

    }
    else
    {
        System.out.println("it's not an Annagram String");
    }


    }

}
```

Q6. Write to implement panagram checking with least inbuilt methods being used?

Ans

```
public class Pangram
{
    public static void main(String[] args)
    {
       boolean flag=false;
        String str="THE QUICK BROWN FOX JUMPS OVER LAZY DOG";
        str=str.replace(" ", "");
        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("Its not pangram");
                flag=true;
            }
        }

        if(flag==false)
        {
            System.out.println("Its pangram");
        }



    }

}
```

Q7. Write a find if string contains all unique characters.

Ans

```
public class countCVS {

    public static void main(String[] args) {
        String str = "HEllo this is java course!!!!!!!!@@@##";
```

```
        str=str.toLowerCase();
        // int num = 0;
        // int sp = 0;
        // int vowels = 0;
        // int constent = 0;
        int specialChars = 0;
        char ch;
        for ( int i = 0; i < str.length(); i++) {
            ch=str.charAt(i);
            if (ch>='a' && ch<='z') {
                System.out.print("");
            }
            else if (ch>='0' &&ch<='9'){
           System.out.println("");
            }
            else if (ch==' '){
            System.out.print("");
            }
            else
            {
                specialChars++;
            }

        }
    //    System.out.println("total number of vowels is: "+vowels);
    //    System.out.println("total number of constents is: "+constent);
    //    System.out.println("total number of space is: "+sp);
    //    System.out.println("total number of numbers is: "+num);
       System.out.println("total number of Spcial characters is: "+specialChars);
    }

}
```

Q7. WAP to find the maximum accurring character in a string.

Ans.
```
public class maximumacrsing {

    public static void main(String[] args) {
        String str = "pw java is the awesome course";
        str=str.replace(" ", "");
        str=str.toLowerCase();
        int arr[] = new int[256];
        for (int i = 0; i < str.length(); i++) {
            arr[str.charAt(i)] = arr[str.charAt(i)] + 1;

        }
        int max = -1;
        char c = ' ';
        for (int i = 0; i < str.length(); i++) {

            if (max < arr[str.charAt(i)]) {

                max = arr[str.charAt(i)];
                c = str.charAt(i);
            }
        }
        System.out.println("maximum repeated charactre is " + c);

    }

}
```