

# Assignment 3

SAPNA  
PAGE NO.:  
DATE: / /

17 How to print duplicate characters from string?

For example, if string is "JAVA" then the program should print "a".

```
public class Example
{
    public static void main (String args[])
    {
        String str = "beautiful beach";
        char [] carray = str.toCharArray();
        System.out.println ("The string is :" + str);
        System.out.println ("duplicate characters
                           in above string are :");
        for (int i=0; i<str.length(); i++)
        {
            for (int j = i+1; j<str.length(); j++)
            {
                if (carray [i] == carray [j])
                {
                    System.out.println (carray [j] + " ");
                    break;
                }
            }
        }
    }
}
```

2] How to check if two strings are anagrams of each other? A simple coding problem based upon string, but could also be used with numbers. You need to write a java program to check if two given strings are anagrams of each other. Two strings are anagrams if they are written using the same exact letters, ignoring space, punctuation, and capitalization. Each letter should have the same count in both strings. For example, the Ann and Mary are an anagram of each other.

```
import java.util.Arrays;
public class AnagramString {
    static void isAnagram (String str1,
                          String str2)
    {
        String s1 = str1.replaceAll ("\\s");
        String s2 = str2.replaceAll ("\\s");
        boolean status = true;
        if (s1.length () != s2.length ())
            status = false;
        else {
            char [] Anays1 = s1.toLowerCase ().to
            char [] Array ()
```

```
char [] Array s1 = s1. toLowerCase () . to
charArray ()
    Array . sort (Array s1);
    Arrays . sort (Arrays s2);
    status = Arrays . equals (Array s1, Array s2)
}
if (status) {
    System . out . println (s1 + "and" + s2 + "are
    anagrams")
} else {
    System . out . println (s1 + "and" + s2 + "are
    not anagrams")
}
}
public static void main (String [ ] args)
{
    isAnagram ("keep", "peek")
}
```

Q.3] How to program to print first non repeated character from string?  
 one of the most common String interview questions :- Find the first non-repeated (unique) character in a given string. For example, if given string is "Morning" then it should print "M". This question demonstrates the efficient use of the hash table data structure.

```

public class firstRepeated {
    public static void main (String args[])
    {
        String str = "Java Programming";
        int index = findFirstRepeated (str);
        if (index != -1) {
            System.out.println ("First Repeated character");
            str.charAt (index) + " found at index" +
            index);
        }
        else {
            System.out.println ("No repeated character
            found");
        }
    }
}

```

private static int findFirstRepeated (

```
String str) {  
    for (int i=0; i<str.length(); i++) {  
        char c = str.charAt(i);  
        for (int j=i+1; j<str.length(); j++)  
        {  
            if (c == str.charAt(j))  
                return j;  
        }  
    }  
    return -1;  
}
```

4] How to reverse string in java using Iteration and Recursion? Your task is to write a program to reverse string in Java without using the StringBuffer class. You also need to provide both Iterative and recursive algorithms for string reversal. You can use other String utility methods e.g. charAt(), toCharArray() or substring() from java.lang.String class.

public class Stringreverseexample

{

    Public static void main (String args[])

{

        String word = "HelloWorld";

        String reverse = new StringBuffer

(word, reverse().toString());

        System.out.println ("Original string");

        System.out.println ("Reversed string : " + reverse, word, reverse);

        word = "wake up";

        reverse = new

        StringBuilder (word).reverse().toString()

        System.out.println ("Original string : ");

        System.out.println ("Reversed string : " + reverse, word, reverse)

```
word = "Band"
reverse = reverse(word)
System.out.println ("original string: " + word)
reversed string: " + n", word, reverse)
}
```

```
public static String reverse
(String source) {
    if (source == null || source.isEmpty())
        return source;
}
```

```
String reverse = "";
for (int i = source.length() - 1; i >= 0; i--) {
    reverse = reverse + source.charAt(i);
}
return reverse
}
```

Q.5) How to find duplicate characters in a string? you need to write a program to print all duplicate character and their count in java. For example, if given string is "Programming" then your program should print.

f : 2

r : 2

m : 2

```
→ import java.util.HashMap;
import java.util.Map;
import java.util.Set;
public class DuplicateCharFinder {
    public void findIt(String str) {
        Map<Character, Integer> baseMap = new
        HashMap<Character, Integer>();
        char[] charArray = str.toCharArray();
        for (Character ch : charArray) {
            if (baseMap.containsKey(ch)) {
                baseMap.put(ch, baseMap.get(ch) + 1);
            } else {
                baseMap.put(ch, 1);
            }
        }
        Set<Character> keys = baseMap.keySet();
```

```
for (Character ch : keys) {  
    if (baseMap.get(ch) >= 1) {  
        System.out.println(ch + " is " + baseMap.get(ch) + " times");  
    }  
}
```

```
public static void main (String args[]) {  
    DuplicatecharFinder dcf = new Duplicatechar  
    Finder();  
    dcf.findIt ("india is my country");  
}
```

Q.8] How to count a number of vowels and consonants in a string? one of the easiest string questions you will ever see. You have to write a java program that will take a string input and print out a number of vowels and consonants on that string.  
 For example, if the input is "java" then your program should print "2 vowels and 2 consonants".

```
public class CountVowelConsonants{
    public static void main (String args[])
    {
        int vCount = 0, cCount = 0;
        String str = "This is a really simple
                    Sentence";
        str = str.toLowerCase();
        for (int i=0; i < str.length(); i++)
        {
            if (str.charAt(i) == 'a' || str.charAt(i) ==
                'e' || str.charAt(i) == 'i' || str.charAt(i) ==
                'o' || str.charAt(i) == 'u')
                vCount++;
            else if (str.charAt(i) >= 'a' & str.charAt(i) <= 'z')
                cCount++;
        }
        System.out.println("Number of Vowels: " + vCount);
        System.out.println("Number of Consonants: " + cCount);
    }
}
```

```
char At (i) <= 'z') {  
    count++;  
}  
System.out.println ("Number of vowels" + v  
count++);  
System.out.println ("Number of constants"  
count);  
}  
}
```

Q.7] How to count the occurrence of a given character in string?

```

→ class javaExample {
    static void countEachChar (String str)
    {
        int counter []= new int [256];
        int len = str.length ();
        for (int i=0 ; i<len ; i++)
            counter [str.charAt (i)]++;
        char array []= new char [str.length ()];
        for (int i=0 ; i<len ; i++)
        {
            array [i] = str.charAt (i);
            int flag=0;
            for (int j=0 ; j<i ; j++)
            {
                if (str.charAt (j) == array [i])
                    flag++;
            }
            if (flag == 1)
                System.out.println (" occurrence of char "
                    + str.charAt (i) + " in the string is :" +
                    counter [str.charAt (i)]);
        }
    }
}

```

String str = "beginner book";

Count Each char (str):

8] How to convert numeric string to an int

```
public class stringToIntExample {  
    public static void main (String args [])  
    {  
        String s = "300";  
        int i = Integer.parseInt (s);  
        System.out.println (i);  
    }  
}
```

Output :

300

Q.9] How to find all permutations of string?

```
public class PermutationString {
    public static void main String swapString
    (String a, int i, int j) {
        char[] b = a.toCharArray();
        char ch;
        ch = b[i];
        b[i] = b[j];
        b[j] = ch;
        return String.valueOf(b);
    }
}
```

```
public static void main (String args)
{
    String str = "ABC";
    int len = str.length();
    System.out.println ("All the permutations
    of the string are:");
    generatePermutation (str, 0, len);
}
```

```
public static void generatePermutation
(String str, int start, int end)
{
    if (start == end - 1)
        System.out.println (str);
    else
```

```

    {
        for (int i = start; i < end; i++)
        {
            str = swapString (str, start, i);
            generatePermutation (str, start, end);
            str = swapString (str, start, i);
        }
    }
}

```

Q. 10) How do check if string is Palindrome

```

public class Palindrome
{
    public static void main (String arg[])
    {
        String a, b;
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the string you
        want to check:");
        a = s.nextLine();
        int n = a.length();
        for (int i = n - 1, j = 0; i >= 0; i--, j++)
        {
            b = b + a.charAt(i);
        }
    }
}

```

```
if (a.equalsIgnoreCase (b))  
{  
    System.out.println ("The string is Palindrome");  
}  
else  
{  
    System.out.println ("The string is not Palindrome");  
}  
}  
}
```

Q13) How to remove duplicate characters from string?

```
import java.util.*;  
class GFG  
{  
    static String removeDuplicate (char str,  
                                  int n)  
    {  
        int index = 0;  
        for (int i = 0; i < n; i++)  
        {  
            int j;  
            for (j = 0; j < i; j++)  
            {  
                if (str[i] == str[j])  
                {  
                    break;  
                }  
            }  
            if (j == i)  
            {  
                str[index] = str[i];  
                index++;  
            }  
        }  
        return str;  
    }  
}
```

break;

}

}

if (j == i)

{

str [index++ ] = str [i];

}

turn string value of (Arrays . copyof (str,  
index));

char str[] = "geeksforgeeks" to charan

( );

int n = str.length;

System.out.println (removeduplicate (str,

)

)

[2] How to return the highest occurred character in a string.

public class GFG

{

```

static final int ASCII_SIZE = 256;
static char getmaxoccurringchar (String str)
{
    int count[] = new int [ASCII_SIZE - SIZE];
    int len = str.length();
    for (int i=0; i<len; i++)
        count [str.charAt(i)]++;
    int max = -1;
    char result = '';
    for (int i=0; i<len; i++)
    {
        if (max < count [str.charAt(i)])
        {
            max = count [str.charAt(i)];
            result = str.charAt(i);
        }
    }
    return result;
}
public static void main (String args[])
{
    String str = "Sample String";
}
```

System.out.println("Max occurring character is:" + getMaxOccurringChar(str));

{}