

Assignment 3

Ques 1. How to print duplicate characters from String? For example, if String is "Java" then program should print "a".

public class Example

```

public static void main (String
args[])
{
    String str = "beautiful beach";
    char [] corray = 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 (corray [i] == corray [j])
            {
                System.out.println (corray [j] + " ");
                break;
            }
        }
    }
}

```

Output:-

The String is : beautiful beach
Duplicate Characters in above string
are : b e a u.

2) How to check if two strings are anagrams of each other?

A simple coding problem based upon string, but could also be asked 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 Army and mary are an anagram of each other.

```
import java.util.Arrays;
public class AnagramString {
    static void isAnagram(String str1,
                          String str2) {
        String st = str1.replaceAll("\n\t\r\b\f");
        String s1 = st.toLowerCase();
        String s2 = str2.toLowerCase();
        char[] arr1 = s1.toCharArray();
        char[] arr2 = s2.toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        if (arr1.length != arr2.length)
            System.out.println("Not an anagram");
        else
            for (int i = 0; i < arr1.length; i++)
                if (arr1[i] != arr2[i])
                    System.out.println("Not an anagram");
                else
                    System.out.println("An anagram");
    }
}
```

```
String s2 = str2.replaceAll(" ", "");  
boolean status = true;  
if (str.length() != s2.length())  
{
```

```
    status = false;
```

```
} else if
```

```
char[] arrayS1 = s1.toLowerCase();  
toCharArray();
```

```
char[] arrayS2 = s2.toLowerCase();  
toCharArray();
```

```
Arrays.sort(arrayS1);
```

```
Arrays.sort(arrayS2);
```

```
status = Arrays.equals(arrayS1,  
arrayS2);
```

```
}
```

```
if (status) {
```

```
System.out.println(str + " and " + s2 +  
" are anagrams");
```

```
} else if
```

```
System.out.println(str + " and " + s2 +  
" are not anagrams");
```

```
}
```

```
}
```

```
public static void main (String args[])
```

```
{
```

```
    isAnagram ("keep", "peek");
```

isAnagram ("motherInLaw", "HitlerWoman");

}

}

Output:-

keep and peek are anagrams
motherInLaw and HitlerWoman are
anagrams.

- 3) How to program to print first non-repeated character from string?
once 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;
    }
}
```

Output:-

First Repeated character a found at index 3.

4. How to reverse string in java using iteration and recursion?

```
public class StringReverseExample  
{  
    public static void main (String args)  
    {  
        String input; // input string  
        String word = "HelloWorld";  
        String reverse = new StringBuffer  
(word).reverse().toString();  
        System.out.println ("original string:  
        %s, reversed string %s %n", word,  
        reverse);  
        word = "Wake UP";  
        reverse = new  
        StringBuilder (word).reverse().toString();  
        System.out.println ("original string:  
        %s, reversed string %s %n", word,  
        reverse);  
        word = "Band";  
        reverse = reverse (word);  
        System.out.println ("original string:  
        %s, reversed string %s %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[i];  
    }  
    return reverse;
```

Output:-

original string :- HelloWorld,

reversed string :- olleHdrow

original string :- Wakeup, reversed
string :- pukeW

original string :- Band, reversed
string :- dnaB

5) How to find duplicate characters in a string?

```
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class Details {
    public void countDupChars(String str) {
        Map<Character, Integer> map = new
        HashMap<Character, Integer>();
        char[] chars = str.toCharArray();
        for (Character ch : chars) {
            if (map.containsKey(ch)) {
                map.put(ch, map.get(ch) + 1);
            } else {
                map.put(ch, 1);
            }
        }
    }
}
```

```
Set<Character> keys = map.keySet();
for (Character ch : keys) {
    if (map.get(ch) > 1) {
        System.out.println("char " + ch + " "
                           + map.get(ch));
    }
}
```

```
Details obj = new Details();
System.out.println("String : Beginner
BOOK.com");
obj.countDuplicates("Beginner BOOK.
com");
System.out.println("inString :
chaitanyaSingh");
obj.countDuplicates("chaitanyaSingh");
}
} // main
```

Output: BeginnersBook.com

char e 2

char B 2

char n 2

char o 3

String : chaitanyaSingh

char a 3

char n 2

char h 2

char i 2

6) HOW TO COUNT A NUMBER OF VOWELS AND CONSONANTS IN A STRING?

```

public class CountVowelConsonant {
    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);
    }
}

```

System.out.println("Number of vowels" + vCount);

System.out.println("Number of consonants!" + cCount);

-3.

}

Output:-

Number of vowels : 10

Number of consonants : 17

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(i) == array[j])  
                flag++;  
        }  
    }  
}
```

```
if (flag == 1) {  
    System.out.println("occurrence of  
    char " + str.charAt(i) + " in the  
    string is :" + counter[stx.charAt  
    (i)]); }  
}
```

```
String str = "beginnersbook";  
CountEachChar(str);  
}
```

Output:

occurrence of char b in the string is : 2
occurrence of char e in the string is : 2
occurrence of char g in the string is : 1
occurrence of char i in the string is : 1
occurrence of char n in the string is : 2
occurrence of char x in the string is : 1
occurrence of char s in the string is : 1
occurrence of char o in the string is : 2
occurrence of char k in the string is : 1

i) How to convert numeric string to
an int?

```
public class StringToIntExample {  
    public static void main (String args []) {  
        }
```

```

String s = "200";
int i = Integer.parseInt(s);
System.out.println(i);
}
}

Output:
200
  
```

Q) How to find all permutations of String?

```

public class permuteString {
    public static 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 + 1, end);  
            str = swapString(str, start, i);  
        }  
}
```

Output:

All the permutations of the string are:

ABC

ACB

BAC

BCA

CBA

CAB

10) How to check if string is palindrome?

Output:-

Enter the string you want to check:
NeveroddorEven

The String is Palindrome.

How to remove duplicate characters from string?

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

```
class OFB
```

```
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];`

}

turn string::valueof(Arrays::copyof
[str, index]);

}

public static void main (String args[])

char str [] = "geeksforgeeks", to
charArray [];

int n = str.length();

System.out.println (removeDuplicate
[str, n]);

}

}

Output:
geeksforgeeks

- 12) How to return the highest occurred character in a string?

public class GFG

{

static final int ASCII_SIZE = 256;

static char getMaxOccuringChar (String str)

int count [] = new int [ASCII_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:" + getMaxOccuringChar
        (str));
}
```

Output:
Max occurring character is: S