**Strings in Java–Assignment**

Question 1: -**Write a program to remove Duplicates from String .(Take any String example with duplicates characters.**

Ans: -

**package** hello;

**import** java.util.Arrays;

**public** **class** RemoveDuplicate {

**static** **void** removeDuplicate(**char** str[], **int** length) {

**int** index = 0;

**for** (**int** i = 0; i < length; i++) {

**int** j;

**for** (j = 0; j < i; j++) {

**if** (str[i] == str[j]) {

**break**;

}

}

**if** (j == i) {

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

}

}

System.***out***.println(String.*valueOf*(Arrays.*copyOf*(str, index)));

}

**public** **static** **void** main(String[] args) {

String course = "PSkill";

**char** str[] = course.toCharArray();

**int** len = str.length;

*removeDuplicate*(str, len);

}

}

Question 2: -**WAP to print Duplicate characters from the String.**

Ans: -**package** hello;

**public** **class** FindDuplicate {

**static** **void** FindDuplicate(**char** str[], **int** length) {

**for** (**int** i = 0; i < length; i++) {

**int** j;

**int** count = 1;

**for** (j = i + 1; j < length; j++) {

**if** (str[i] == str[j] && str[i] != ' ') {

count++;

str[j] = ' ';

}

}

**if** (count > 1)

System.***out***.println(str[i]);

}

}

**public** **static** **void** main(String[] args) {

String course = "this is PWSkill aa";

**char** str[] = course.toCharArray();

**int** len = str.length;

FindDuplicate (str, len);

}

}

Question 3: -**WAP to check if “2552” is palindrome or not?.**

Ans: -

**package** hello;

**public** **class** CheckPelindrome {

**public** **static** **void** main(String[] args) {

String str="2552";

String str1="";

**char**[]str2=str.toCharArray();

//Object value;

**for**(**int** i=str2.length-1;i>=0;i--) {

str1=str1+str2[i];

}

System.***out***.print(str1);

**if**(str==str1) {

System.***out***.println("String is palindrome");

}

}

}

Question 4: -**WAP to count the number of constraints ,vowels,special characters in a String.**

Ans: -

**package** hello;

**public** **class** counttype {

**static** **void** countCharacterType(String str) {

**int** vowels = 0, consonant = 0, specialChar = 0, digit = 0;

**for** (**int** i = 0; i < str.length(); i++) {

**char** ch = str.charAt(i);

**if** ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {

ch = Character.*toLowerCase*(ch);

**if** (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')

vowels++;

**else**

consonant++;

} **else** **if** (ch >= '0' && ch <= '9')

digit++;

**else**

specialChar++;

}

System.***out***.println("Total Vowels:= " + vowels);

System.***out***.println("Total Consonant:=" + consonant);

System.***out***.println("Total Digit:=" + digit);

System.***out***.println("Total Special Character:=" + specialChar);

}

**public** **static** **void** main(String[] args) {

String str = "My name is Vikash chauhan 123321@#";

*countCharacterType*(str);

}

}

Question 5: -**WAP to implement Anagram checking least inbuilt methods being used.**

Ans: -

**package** hello;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** AnagramCheck {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter first string");

String str1=sc.nextLine();//school master

System.***out***.println("Enter Second string");

String str2=sc.nextLine();//the class room

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's anagram");

}

**else** {

System.***out***.println("it's now anagram");

}

}

}

Question 6: -**WAP to implement Pangram checking with least inbuilt methods being used.**

Ans: -

**package** hello;

**public** **class** PangramCheck {

**public** **static** **void** main(String[] args) {

**boolean** con = **false**;

String str = "The quick BRown fox jumps over lazy dog";

str = str.replace(" ", "");

str = str.toUpperCase();// 97

**char**[] arr = str.toCharArray();

**int**[] arr1 = **new** **int**[26];

**int** count = 0;

**for** (**int** i = 0; i < arr.length; i++) {

arr1[arr[i] - 65]++;

}

**for** (**int** i = 0; i < arr1.length; i++) {

**if** (arr1[i] == 0) {

System.***out***.println("It's not Pangram");

con = **true**;

}

}

**if** (con == **false**) {

System.***out***.println("it is pangram");

}

}

}

Question 7: -**WAP to find if string contains all unique characters.**

Ans: -

**package** hello;

**public** **class** uniqueString {

**boolean** uniqueCharacters(String str) {

**for** (**int** i = 0; i < str.length(); i++) {

**for** (**int** j = i + 1; j < str.length(); j++) {

**if** (str.charAt(i) == str.charAt(j)) {

**return** **false**;

}

}

}

**return** **true**;

}

**public** **static** **void** main(String args[]) {

uniqueString obj = **new** uniqueString();

String name = "vikash";

**if** (obj.uniqueCharacters(name)==**true**)

System.***out***.println("The String :+" + name + "+: has all unique characters");

**else**

System.***out***.println("The String " + name + " has duplicate characters");

}

}

Question 8: -**WAP to find the maximum occurring characters in a string.**

Ans: -

**package hello;**

**public class FindMaxreppitCharacters {**

**static final int *N* = 256;**

**static char MaxOccuringChar(String str1) {**

**int character[] = new int[*N*];**

**for (int i = 0; i < str1.length(); i++) {**

**character[str1.charAt(i)]++;**

**}**

**int max = -1;**

**char result = ' ';**

**for (int i = 0; i < str1.length(); i++) {**

**if (max < character[str1.charAt(i)]) {**

**max = character[str1.charAt(i)];**

**result = str1.charAt(i);**

**}**

**}**

**return result;**

**}**

**public static void main(String[] args) {**

**String str1 = "Hello PWskill";**

**System.*out*.println("Max occurring character in the given string is:=" + *MaxOccuringChar*(str1));**

**}**

**}**