**Assignment3(Nitin Jain)**

1. WAP to reverse a String.

Input: “iNeuron”

Output: “norueNi”

***Program:***

public class Question\_1 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the String");

String input=sc.nextLine();

char inputCharArr[]=input.toCharArray();

for(int i=inputCharArr.length-1;i>=0;i--)

System.out.print(inputCharArr[i]);

}

}

***Output:-***

A picture containing text

Description automatically generated

2. WAP to reverse a sentence while preserving the position.

Input : “Think Twice”

Output : “kniht eciwt”

***Program:***

public class Question\_2 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the String");

String input=sc.nextLine().toLowerCase();

String inputSplitArr[]=input.split(" ");

char particularString[];

for(int i=0;i<inputSplitArr.length;i++)

{

particularString=inputSplitArr[i].toCharArray();

for(int j=particularString.length-1;j>=0;j--)

{

System.out.print(particularString[j]);

}

System.out.print(" ");

} }

}

***Output:-***

A picture containing shape

Description automatically generated

3. WAP to check if the String is Anagram or not.

***Program:-***

public class Question\_3 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

System.out.println("Enter the Second Strings");

String input2 = sc.nextLine().toLowerCase();

if(input1.length()==input2.length())

{

char [] input1CharArr=input1.toCharArray();

char [] input2CharArr=input2.toCharArray();

Arrays.sort(input1CharArr);

Arrays.sort(input2CharArr);

int flag=0;

for(int i=0;i<input1CharArr.length;i++)

{

if(input1CharArr[i]!=' ' && input2CharArr[i]!=' ')

{

if(!(input1CharArr[i]==input2CharArr[i]))

flag=1;

}

}

if(flag==0)

System.out.println("Both Strings Are Anagram");

else

System.out.println("Both Strings Are Not Anagram");

}

else

System.out.println("Both Strings Are Not Anagram");

}

}

***Output:-***

A picture containing table

Description automatically generated

4. WAP to check if the String is a Pangram or not.

***Program:-***

public class Question\_4 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

if(input1.length()>=26)

{

char arr[] = input1.toCharArray();

Arrays.sort(arr);

int flag = 0, counter = 0, asciiCounter = 97;

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

if (arr[i] == asciiCounter && counter <= 25 && arr[i] >= 97 && arr[i] <= 122) {

counter++;

asciiCounter++;

} else if (arr[i] > asciiCounter) {

flag = 1;

break;

}

}

if (flag == 0 && counter >= 25 && asciiCounter>=122) {

System.out.println("Its Pangram");

} else {

System.out.println("Not Pangram");

}

}

else System.out.println("Not Pangram");

}

}

***Output:***

Graphical user interface, text, application, email

Description automatically generated

5. WAP to print repeatedly occurring characters in the given String.

***Program:-***

public class Question\_6 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

char InputCharArray[]=input1.toCharArray();

Arrays.sort(InputCharArray);

char FirstTimeChar[]=new char[input1.length()];

int FirstTimeCharIndex=0;

char AlreadyPrinted[]=new char[input1.length()];

int AlreadyPrintedIndex=0;

int flag;

StringBuilder RepeatedChars=new StringBuilder();

for(int i=0;i<InputCharArray.length;i++)

{

flag=0;

for(int j=0;j<FirstTimeChar.length;j++)

{

if(InputCharArray[i]==FirstTimeChar[j])

{

flag=1;

break;

}

}

if(flag==0)

{

FirstTimeChar[FirstTimeCharIndex]=InputCharArray[i];

FirstTimeCharIndex++;

}

else

{

flag=0;

for(int j=0;j<AlreadyPrinted.length;j++)

{

if(InputCharArray[i]==AlreadyPrinted[j])

{

flag=1;

break;

}

}

if(flag==0)

{

AlreadyPrinted[AlreadyPrintedIndex]=InputCharArray[i];

AlreadyPrintedIndex++;

RepeatedChars.append(InputCharArray[i]);

} } }

System.out.println(RepeatedChars); }}

***Output:-***

Shape

Description automatically generated

6. WAP to sort a String Alphabetically.

***Program:-***

Class Question\_6{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

char Input1CharArr[]=input1.toCharArray();

Arrays.sort(Input1CharArr);

System.out.println("After Sorting");

for(char c:Input1CharArr)

{

System.out.print(c);

}

}

}

***Output:-:***

Graphical user interface, text

Description automatically generated

7. WAP to count the number of Vowels and Consonants of a String value.

public class Question\_8 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

char Input1CharArr[] = input1.toCharArray();

int VowelsCounter = 0, ConsonentCounter = 0;

for (int i = 0; i < Input1CharArr.length; i++) {

if (Input1CharArr[i] == 'a' || Input1CharArr[i] == 'e' || Input1CharArr[i] == 'i' || Input1CharArr[i] == 'o' || Input1CharArr[i] == 'u') {

VowelsCounter++;

} else if ((Input1CharArr[i] >= 97 && Input1CharArr[i] <= 122) && !(Input1CharArr[i] == 'a' || Input1CharArr[i] == 'e' || Input1CharArr[i] == 'i' || Input1CharArr[i] == 'o' || Input1CharArr[i] == 'u')) { ConsonentCounter++; } }

System.out.println("Vowels-:" + VowelsCounter + "\n" + "Consonent-: " + ConsonentCounter); }}

***Output:-***

Letter

Description automatically generated with medium confidence

8. WAP to count number of special characters.

***Program:-***

public class Question\_9 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the First Strings");

String input1 = sc.nextLine().toLowerCase();

char Input1CharArr[] = input1.toCharArray();

int SpecialCharCounter = 0;

for (int i = 0; i < Input1CharArr.length; i++) {

if ((Input1CharArr[i] >= 32 && Input1CharArr[i] <= 47) || (Input1CharArr[i] >= 58 && Input1CharArr[i] <= 64) || (Input1CharArr[i] >= 91 && Input1CharArr[i] <= 96) || (Input1CharArr[i] >= 123 && Input1CharArr[i] <= 126)) {

SpecialCharCounter++;

}

}

System.out.println("Special characters :" + SpecialCharCounter );

}

}

***Output:-***

