1. **Find the number of occurrences of given word in a sentence.**

static int check(String sentence,String word){

package com.psl.learning.strings;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class WordOccurence {

static int check(String sentence,String word){

int count=0;

String words[] = sentence.split(" ");

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

if (word.equals(words[i]))

count++;

}

return count;

}

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter Sentence and word:");

String sentence=br.readLine();

String word=br.readLine();

System.out.println(word+" occurred "+check(sentence,word)+" Times");

}

}

1. **Ezee Shop has an online portal where customers can check about different products. Customers can search for products by giving a key word. Write a program that accepts a product name and a key word, and checks if the key word is present in the given product name.**

static String[] initProductNames()

static boolean isPresent(String[] productNames, String keyword)

package com.psl.learning.strings;

import java.util.Scanner;

public class ShopSearch {

static String[] initProductNames() {

Scanner in=new Scanner(System.in);

System.out.println("Enter Number of products in the store:");

int number=in.nextInt();

String[] productNames = new String[number];

System.out.println("Enter products in store:");

for(int i=0;i<number;i++)

{

productNames[i]=in.next();

}

return productNames;

}

static boolean isPresent(String[] productNames, String keyword) {

boolean result=false;

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

if(productNames[i].contains(keyword))

result=true;

}

return result;

}

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

String[] productNames=initProductNames();

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

System.out.println(productNames[i]);

System.out.println("Enter the keword:");

String keyword=in.next();

if(isPresent(productNames,keyword))

System.out.println(keyword+" is present in products");

else

System.out.println(keyword+"is not present in products");

in.close();

}

}

1. **Write a program to find the availability of given number in the list.**

static int findPosition(int num,int[] nos)

1. Write a program that will reverse the sequence of letters in each word of a chosen paragraph . For instance, " To be or not to be " Would become " oT eb ro ton ot eb"

static String makeReverse(String str)

package com.psl.learning.arrays;

import java.util.Scanner;

public class ReverseCase {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

System.out.println("Enter the string:");

String str=in.nextLine();

String result=makeReverse(str);

System.out.println("Result:"+result);

}

private static String makeReverse(String str) {

String result="";

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

{

char ch=str.charAt(i);

if(Character.isLowerCase(ch))

result=result+Character.toUpperCase(ch);

else if(Character.isUpperCase(ch))

result=result+Character.toLowerCase(ch);

else

result=result+ch;

}

return result;

}

}

1. Write a function find and replace which will replace the given string from the source string.

package com.psl.learning.strings;

import java.util.Scanner;

public class StringReplace {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

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

String str=in.nextLine();

System.out.println("Enter the string need to replace:");

String existingString=in.next();

System.out.println("Enter the string for replacing:");

String replaceString=in.next();

String result=findAndReplace(str,existingString,replaceString);

System.out.println("Result:"+result);

// TODO Auto-generated method stub

}

private static String findAndReplace(String str, String existingString, String replaceString) {

// TODO Auto-generated method stub

return str.replaceAll(existingString, replaceString);

}

}

1. Write a function findPrime which will accept range of values from user and return int array from function and print all numbers from it

package com.psl.learning.arrays;

import java.util.Scanner;

public class PrimeNumbers {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

int range1=in.nextInt();

int range2=in.nextInt();

int result[]=findPrime(range1,range2);

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

System.out.println(result[i]);

}

private static int[] findPrime(int range1, int range2) {

int result[],k=0;

int number=numberOfPrimes(range1,range2);

result=new int[number];

for(int i=range1;i<=range2;i++) {

if (i == 1 || i == 0)

continue;

int count=0;

for(int j=2;j<=i/2;j++) {

if(i%j==0) {

count++;

break;

}

}

if(count==0)

{

result[k++]=i;

}

}

return result;

}

private static int numberOfPrimes(int range1, int range2) {

int number = 0;

for(int i=range1;i<=range2;i++) {

if (i == 1 || i == 0)

continue;

int count=0;

for(int j=2;j<=i/2;j++) {

if(i%j==0) {

count++;

break;

}

}

if(count==0)

{

number++;

}

}

return number;

}

}

1. Write a function to find various combinations of entered string

static String[] getCombinations(String str)

For Eg str= “123”

Output :

123

132

213

231

312

321

package com.psl.learning.strings;

import java.util.Scanner;

public class StringPermutation {

static String result[];

static int k=0;

static String[] getCombinations(String str) {

int n=str.length();

permute(str,0,n);

return result;

}

private static void permute(String str, int left, int right) {

// TODO Auto-generated method stub

if (left == right)

result[k++]=str;

else

{

for (int i = left; i <= right; i++)

{

str = swap(str,left,i);

permute(str, left+1, right);

str = swap(str,left,i);

}

}

}

public static String swap(String a, int i, int j)

{

char temp;

char[] charArray = a.toCharArray();

temp = charArray[i] ;

charArray[i] = charArray[j];

charArray[j] = temp;

return String.valueOf(charArray);

}

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

String str=in.next();

String result[]=getCombinations(str);

for(int i=0;i<k;i++)

{

System.out.println(result[k]);

}

}

}

1. Create a two dimensional character array which will accept only two characters X or O. populate array with different combinations of X and O characters. If Same character appears at either diagonal position or in the same line, return that character.

For eg.

**X** O O

O **X** O

O O **X**

**X** appears at diagonal position hence **return X**.