DSA PROBLEM PRACTICE DAY 3

PROBLEM 1:

ANAGRAM PROGRAM

CODE;

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the String1 :");

String S1 = scan.next();

System.out.println("Enter the String2 :");

String S2 = scan.next();

char[] ch = S1.toCharArray();

char[] ch2 = S2.toCharArray();

Arrays.sort(ch);

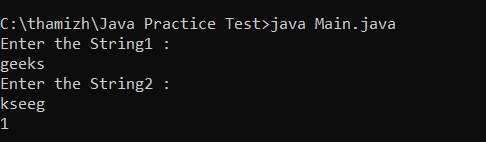
Arrays.sort(ch2);

System.out.println(Arrays.equals(ch,ch2)?1:0);

}

}

OUTPUT :



PROBLEM 2 :

**row with max 1s'**

**CODE :**

**import java.util.\*;**

**public class Main {**

**public static void main(String... argv) {**

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

**System.out.println("Enter the number of rows :");**

**int n = scan.nextInt();**

**System.out.println("Enter the number of cols :");**

**int m = scan.nextInt();**

**int[][] arr = new int[n][m];**

**System.out.println("Enter the Elements 0 or 1 :");**

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

**for(int j=0;j<m;j++){**

**arr[i][j] = scan.nextInt();**

**}**

**}**

**int result = -1;**

**int maxCount = 0;**

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

**int count = helper(arr[i]);**

**if (count > maxCount) {**

**maxCount = count;**

**result = i;**

**}**

**}**

**System.out.println("Index is :" + result);**

**}**

**public static int helper(int[] arr) {**

**int n = arr.length;**

**int low = 0;**

**int high = n - 1;**

**while (low <= high) {**

**int mid = low + (high - low) / 2;**

**if (arr[mid] == 1) {**

**high = mid - 1;**

**} else {**

**low = mid + 1;**

**}**

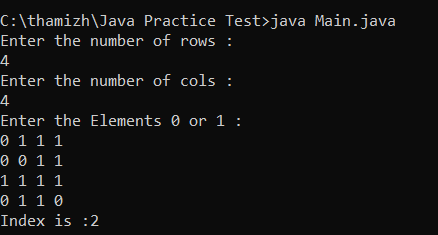
**}**

**return n - low;**

**}**

**}**

**OUTPUT :**

****

PROBLEM 3 :

LONGEST CONSECUTIVE SUBSEQUENCE

CODE :   
import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the size of array :");

int n = scan.nextInt();

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

int[] arr = new int[n];

for(int j=0;j<n;j++){

arr[j] = scan.nextInt();

}

Arrays.sort(arr);

int maxlen=1;

int len=1;

int prev = arr[0];

for(int i=1;i<n;i++){

if(prev == arr[i]){

continue;

}else if(prev == (arr[i]-1)){

len++;

}else{

len=1;

}

prev = arr[i];

maxlen = Math.max(maxlen,len);

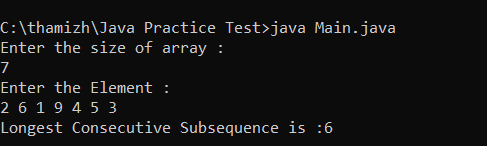
}

System.out.println("Longest Consecutive Subsequence is :" + maxlen);

}

}

OUTPUT :



PROBLEM 4 :

longest palindrome in a string

CODE :

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

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

String s = scan.next();

int n = s.length();

int start = 0;

int maxLen = 1;

for(int i=0;i<n;i++){

int len1 = helper(s,i,i);

int len2 = helper(s,i,i+1);

int len = Math.max(len1,len2);

if(maxLen < len){

maxLen = len;

start = i - (len - 1)/2;

}

}

System.out.println ("Result is :" + s.substring(start,start + maxLen));

}

public static int helper(String s,int left,int right){

while(left>=0 && right<s.length() && s.charAt(left) == s.charAt(right)){

left--;

right++;

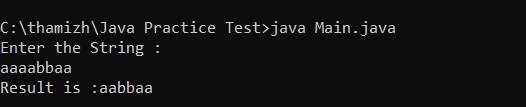
}

return right - left - 1;

}

}

OUTPUT :



PROBLEM 5:

rat in a maze problem

CODE :

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Number of rows :");

int n = scan.nextInt();

System.out.println("Enter the Number of Columns :");

int m1 = scan.nextInt();

int[][] m = new int[n][m1];

System.out.println("Enter the Elements in the matrix 0 or 1 :");

for(int i=0;i<n;i++){

for(int j=0;j<m1;j++){

m[i][j] = scan.nextInt();

}

}

ArrayList<String> result = new ArrayList<>();

boolean[][] board = new boolean[n][n];

if(m[0][0]==0){

result.add("-1");

System.out.println(result);

}else{

helper(0,0,m,board,n,"",result);

System.out.println(result);

}

}

public static void helper(int row,int col,int[][] m,boolean[][] board,int n,String s,ArrayList<String> result){

if(row<0 || row>=n || col<0 || col>=n || board[row][col] || m[row][col]!=1) return;

if(row==n-1 && col==n-1 && m[row][col]==1){

result.add(s);

return;

}

board[row][col] = true;

helper(row+1,col,m,board,n,s+"D",result);

helper(row-1,col,m,board,n,s+"U",result);

helper(row,col+1,m,board,n,s+"R",result);

helper(row,col-1,m,board,n,s+"L",result);

board[row][col] = false;

}

}

OUTPUT:

