**SET A CORE JAVA (CSL5303) ST3**

**1 mark MCQs:**

1) Which of the following manages a list of database drivers in JDBC?

1. **DriverManager**
2. [JDBC driver](javascript:void(0);)
3. [Connection](javascript:void(0);)
4. [Statement](javascript:void(0);)

2) Which of the following is first step to create a JDBC application?

1. **Import packages containing the JDBC classes needed for database programming.**
2. Register the JDBC driver, so that you can open a communications channel with the database.
3. Open a connection using the DriverManager.getConnection () method.
4. Execute a query using an object of type Statement.

3) Method on ResultSet that tests whether or not there remains at least one unfetched tuple in ResultSet, is said to be fetch() method

1. current() method
2. **next() method**
3. access() method

4) Which of the following describes the correct sequence of the steps involved in making a connection with a database?

1. Loading the driver

2. Process the results.

3. Making the connection with the database.

4. Executing the SQL statements.

1. **1,3,4,2**
2. 1,2,3,4
3. 2,1,3,4
4. 4,1,2,3

5) What statements are correct about JDBC transactions?

1. A transaction is a set of successfully executed statements in the database
2. A transaction is finished when commit() or rollback() is called on the Connection object
3. A transaction is finished when commit() or rollback() is called on the Transaction object
4. **A transaction is finished when close() is called on the Connection object.**

6) The …………………… compares the characters inside a sting object whereas ………………. compares two objects references to see whether they refer to the same instance.

1. = = operator , equals( ) method
2. **equals( ) method , = = operator**
3. equals( ) method, = = = operator
4. = = operator, compare( ) method

7) What will be the output of following program?

public class A{

    public A()    {

        super();

        this(10);

    }

    public A(int i)    {

        System.out.println(i);

    }

}

1. **Compile time error, A constructor can have either super() or this() but not both.**
2. Run time error, A constructor can have either super() or this() but not both.
3. 10
4. No output

8) What is the output of this program?

class Output {

public static void main(String args[]) {

char a[] = {'a', '5', 'A', ' '};

System.out.print(Character.isDigit(a[0]) + " ");

System.out.print(Character.isWhitespace(a[3]) + " ");

System.out.print(Character.isUpperCase(a[2]));

}

}

1. true false true
2. **false true true**
3. true true false
4. false false false

9) What is the output of this program?

public class San {

public static void main(String args[]) {

try {

System.out.print("Hello world ");

}

finally {

System.out.println("Finally executing ");

}

}

}

1. The program will not compile because no exceptions are specified
2. The program will not compile because no catch clauses are specified
3. Hello world
4. **Hello world Finally executing**

10) What will be the output of below given program?

class A{

int i;

}

class B extends A{

int j;

void display() {

super.i=j+1;

System.out.println(j+" "+i);

}

}

public class inheritance{

public static void main(String args[]) {

B obj=new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

1. 2 2
2. 3 3
3. **2 3**
4. 3 2

**2 Marks MCQs:**

11) What will be output of following code?

class MyRunnable implements Runnable{

public void run(){

for(int i=0;i<3;i++){

System.out.println ("i="+i);

}

}

}

public class mcq{

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

System.out.println("In main() method");

MyRunnable runnable =new MyRunnable();

Thread th1=new Thread(runnable);

Thread th2=new Thread(runnable);

th1.start();

th1.join();

th2.start();

th2.join();

System.out.println("end main() method");

}

}

a.In main() method  
        i=0   
        i=1   
        i=0   
        i=2   
        i=1   
        end main() method  
        i=2

b. In main() method  
        i=0   
        i=1   
        i=0   
        i=2   
        i=1   
        i=2   
        end main() method

c**. In main() method  
        i=0   
        i=1   
        i=2**

**i=0   
        i=1   
        i=2   
        end main() method**

d. None of these

12) What will be the output of following code?

class ClassOne{

    void method(String s1)    {

        method(s1, s1+s1);

    }

    void method(String s1, String s2)    {

        method(s1, s2, s1+s2);

    }

    void method(String s1, String s2, String s3)    {

        System.out.println(s1+s2+s3);

    }

}

public class MainClass{

    public static void main(String[] args)    {

        ClassOne one = new ClassOne();

        one.method("JAVA");

    }

}

1. **JAVAJAVAJAVAJAVAJAVAJAVA**
2. JAVAJAVAJAVAJAVAJAVA
3. JAVAJAVAJAVAJAVA
4. JAVAJAVAJAVA

13) What will be the output of below given program?

class A{

    int[] a = new int[5];    {

        a[0] = 10;

    }

}

public class MainClass extends A{

    {

        a = new int[5];

    }

    {

        System.out.println(a[0]);

    }

    public static void main(String[] args)    {

        MainClass main = new MainClass();

    }

}

1. 10
2. **0**
3. NULL
4. Undefined

14) What will be the output of following program?

import java.io.\*;

public class Chararrayinput {

public static void main(String[] args) {

String obj = "abcdef";

int length = obj.length();

char c[] = new char[length];

obj.getChars(0, length, c, 0);

CharArrayReader input1 = new CharArrayReader(c);

CharArrayReader input2 = new CharArrayReader(c, 0, 3);

int i;

try {

while ((i = input2.read()) != -1) {

System.out.print((char)i);

}

}

catch (IOException e) {

e.printStackTrace();

}

}

}

1. **abc**
2. abcd
3. abcde
4. abcdef

15) What is the output of given code?

class A{

public int i;

protected int j;

}

class B extends A{

int j;

void display() {

super.j=3;

System.out.println(i+" "+j);

}

}

public class output{

public static void main(String args[]) {

B obj=new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

1. **1 2**
2. 2 1
3. 1 3
4. 3 1

**16) *COMPILER***

Maera is now very keen in compiler development. She knows that one of the most important components of a compiler is its parser. A parser is a software component that processes text, and checks its semantic correctness or if the text is properly built.

Today, she is very concerned with an abstract instruction which is composed of the characters ‘<’ and ‘>’, which she will use in the design of her language, L++.

She is using it as an abstraction for generating XML code tags in an easier fashion and she understood that, for an expression to be valid, a ‘<’ symbol must always have a corresponding ‘>’ character somewhere (not necessary immediately) after it. Moreover, each ‘>’ symbol should correspond to exactly one ‘<’ symbol.

So, for instance, the instructions:

<<>>

<>

<><>

are all valid. While:

>>

><><

are not.

Given some expressions which represent some instructions to be analyzed by Maera’s compiler, you should tell the length of the longest prefix of each of these expressions that is valid, or 0 if there's no such a prefix.

*Input Format*

Input will consist of an integer T denoting the number of test cases to follow.

Then, T strings follow, each on a single line, representing a possible expression in L++.

*Output Format*

For each expression you should output the length of the longest prefix that is valid or 0 if there's no such a prefix.

*Sample Input 1*

3

<<>>

><

<>>>

*Sample Output 1*

4

0

2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2  <<>>  <>>> | 4  <<>>  <>>>  <<<>>>  <<><><> | 3  <<<<>>>  <<><>>  <><><><><><><><><><> | 6  <<><><<>  <<>><><>  <<><><<<>>>><  >>>>>>><<<<<<<<>>>>>>>>>  <<>><><><><>  <<<>>>><<<>>>>><><><> | 20  <<<><>>  <><><>  <<<>>>  <<><<><>>  <><<<>>>  <<<><>>  <><><>  <<<>>>  <<><<><>>  <><<<>>>  <<<><>>  <><><>  <<<>>>  <<><<><>>  <><<<>>>  <<<><>>  <><><>  <<<>>>  <<><<><>>  <><<<>>> |
| 4  2 | 4  2  6  0 | 0  6  20 | 0  8  12  0  12  6 | 0  6  6  0  8  0  6  6  0  8  0  6  6  0  8  0  6  6  0  8 |

*Solution*:

import java.util.\*;

import java.lang.\*;

import java.io.\*;

public class file{

public static void main (String[] args) throws java.lang.Exception {

Scanner sc=new Scanner(System.in);

int testCases=sc.nextInt();

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

String name=sc.next();

char arr[]=name.toCharArray();

int c=0;

Stack<Character> stack=new Stack<Character>();

for(int i1=0;i1<arr.length;i1++){

char ch=arr[i1];

if(ch=='<')

stack.push(ch);

else if(ch=='>'&& stack.empty()==false )

stack.pop();

else

break;

if(stack.empty())

c = i1;

}

if(c==0)

System.out.println(c);

else

System.out.println(c+1);

}

}

}

17) ***Heads in an array***

Write a program to print all the HEADS in the array. An element is a head if it is greater than all the elements to its right side. And the rightmost element is always a head.

For example in the array {16, 17, 4, 3, 5, 2}, heads are 17, 5 and 2.

*Input Format*

First line of each test contains an integer N denoting the size of the array.

The next N lines denote the elements of the array.

*Output Format*

Display all the HEADS in the array

*Sample Input 1*

5

23

14

34

20

19

*Sample Output 1*

34

20

19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6  24  12  26  13  14  11 | 5  55  44  33  22  11 | 4  123456  234567  210000  222222 | 7  1  0  0  0  9  8  7 | 2  9999999  8888888 |
| 26  14  11 | 55  44  33  22  11 | 234567  222222 | 9  8  7 | 9999999  8888888 |

***Solution***

import java.util.\*;

import java.lang.\*;

import java.io.\*;

public class file {

static void printHeads(int arr[], int size)

{

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

**int j;**

**for (j = i + 1; j < size; j++)**

**{**

**if (arr[i] <= arr[j])**

**break;**

**}**

**if (j == size) // the loop didn't break**

**System.out.print(arr[i] + " ");**

**}**

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int arr[] = new int[n];

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

arr[i] = sc.nextInt();

printHeads(arr, n);

}

}

18) ***Complete the word***

Detective Bakshi while solving a case stumbled upon a letter which had many words whose one character was missing i.e. one character in the word was replaced by an underscore. For example: "Fi\_er". He also found thin strips of paper which had a group of words separated by colons, for e.g. "Fever:filer:Filter Fixer:fiber:fibre:tailor:offer". He could figure out that the word whose one character was missing was one of the possible words from the then thin strips of paper. Detective Bakshi has approached you (a computer programmer) asking for help in identifying the possible words for each incomplete word.

You are expected to write a function to identify the set of possible words. The function Identify Possible Words takes two strings as input where, input1 contains the incomplete Word and input2 is the string containing set of words separated by colons. The function is expected to find all the possible words from input2 that can replace the incomplete word in input1, and return the result in the string form in which matched words are in upper case, separated by colon and added to result string in the order in which they appeared in lnput2.

If none of the words in input2 are possible candidates to replace input1, the output string should contain the string "ERROR-009"

*Input Format*

The first line of the input contains an integer T, denoting the number of test cases.

For each test, next two lines are the two strings as input where, input1 contains the incomplete Word, and input2 is the string containing a set of words separated by colons.

*Output Format*

String should be returned as all words in upper case and the output string should appear in the order they matched with input 2 and separated by colons

If none of the words in input2 are possible candidates to replace input1, the output string should contain the string "ERROR-009"

*Sample Input 1*

2

Fi\_er

Fever:filer:Filler:Fixer:fiber:fibre:tailor:offer

del\_te

deleted:deleting

*Sample Output 1*

*FILER:FIXER:FIBER*

*ERROR-009*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1  Fi\_er  Fever:filer:fixer:fiber:fibre:tailor:offer:fi1er | 2  Cod\_  Code:codes  Ou\_put  Output:Output:Output | 3  \_ode  Code:codes  sav\_  saved:saving:saved:saving:saved:saving:saved:saving:save:save:save:sav#:sav$  chi\_kara  chitkara:chitkara | 4  Custo\_  custom:custor:custos:cpstom  r\_n  running:run:run:running:run  cann\_t  can:can:cant:cant  diffi\_ult  difficulty:difficulty | 5  \_ustom  custom:mustom:gustom:cpstom  ente\_  entered:entering:entered:entered:entering:entered:entered:entering:entered:entered:entering:entered:entered:entering:entered  word\_  words:word&:worde:wording  \_ave  saved:saving:saving:save:cave:pave:jave:tave:kave:save:kave  vi\_eo  vvvv:vid:dhdjdj |
| FILER:FIXER:FIBER:FI1ER | CODE  OUTPUT:OUTPUT:OUTPUT | CODE  SAVE:SAVE:SAVE:SAV#:SAV$  CHITKARA:CHITKARA | CUSTOM:CUSTOR:CUSTOS  RUN:RUN:RUN  ERROR-009  ERROR-009 | CUSTOM:MUSTOM:GUSTOM  ERROR-009  WORDS:WORD&:WORDE  SAVE:CAVE:PAVE:JAVE:TAVE:KAVE:SAVE:KAVE  ERROR-009 |

***Solution:***

import java.io.\*;

import java.util.\*;

import java.util.regex.\*;

public class file

{

static String identifyPossibleWords(String input1,String input2){

**input1=input1.toUpperCase();**

**input2=input2.toUpperCase();**

**String str[]=input2.split(":");**

**int n=input1.length();**

**int x=input1.indexOf('\_');**

**char match[]=input1.toCharArray();**

**String res="";**

**for(String s:str){**

**char match1[]=s.toCharArray();**

**if(n==s.length()){**

**match1[x]='\_';**

**if(Arrays.equals(match,match1))**

**{**

**if(res.equals(""))**

**res=res+s;**

**else**

**res=res+":"+s;**

**}**

**}**

**}**

**if(res.equals(""))**

**res="ERROR-009";**

**return res;**

}

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

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

int t=Integer.parseInt(reader.readLine());

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

String input1=reader.readLine();

String input2=reader.readLine();

String res=identifyPossibleWords(input1,input2);

System.out.println(res);

}

}

}