IRC_JAVA_COD_Strings_Set2

Test Summary

- No. of Sections: 1
- No. of Questions: 10
- Total Duration: 180 min

Section 1 - Coding

Section Summary

- No. of Questions: 10
- Duration: 180 min

Additional Instructions:

None

Q1. **Using contains() and trim() method**

Having finished most of our application for the fair, it's time to focus on minor details that went wrong during a test run of our application in this module. Accidentally some gibberish text with leading and trailing got copied to the clipboard and got pasted in the some of your text documents. Don't worry, still, we have the gibberish text with us, you can manually load each document, and find the text and delete it. Think it will take ages, no we can think of a time saver. Using your programming skills, load each document in a program and find in which files the text got copied. Assume text of the document is given as the input to the program. write a program to find whether the gibberish text is present in the string.

Create a driver class called Main. In the Main method, obtain the inputs from the console (Refer I/O) and prompt whether the gibberish text is present in the main text.

Input Format

First line of the input consist of a sentence
Second line of the input consist of a string in the sentence

Output Format

Output prints whether the string is found in the sentence or not

Sample Input

```
One fine morning, a minister from Emperor Akbar's cour
stolen
```

Sample Output

```
String is found in the sentence
```

Sample Input

```
One fine morning, a minister from Emperor Akbar's cour
 stolen
```

Sample Output

```
String is found in the sentence
```

Sample Input

```
One fine morning, a minister from Emperor Akbar's cour
 account
```

Sample Output

```
String is not found in the sentence
```

Sample Input

```
One fine morning, a minister from Emperor Akbar's cour
account
```

Sample Output

```
String is not found in the sentence
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. **Alternating Code**

It is IPL Season and the first league match of Dhilip’s favorite team, "Chennai Super Kings". The CSK team is playing at the IPL after 2 years and like all Dhoni lovers, Dhilip is also eagerly awaiting to see Dhoni back in action.
After waiting in long queues, Dhilip succeeded in getting the tickets for the big match. On the ticket, there is a letter-code that can be represented as a string of upper-case Latin letters.
Dhilip believes that the CSK Team will win the match in case exactly two different letters in the code alternate. Otherwise, he believes that the team might lose. Please see note section for formal definition of alternating code.

You are given a ticket code. Please determine, whether CSK Team will win the match or not based on Dhilip'sconviction. Print "YES" or "NO" (without quotes) corresponding to the situation.
Note:
Two letters x, y where x != y are said to be alternating in a code, if code is of form "xyxyxy...".

Input Format

First and only line of the input contains a string S denoting the letter code on the ticket.

Output Format

Output a single line containing "Yes" (without quotes) based on the conditions given and "No" otherwise.
Refer sample input and output for formatting specifications.

Sample Input

```
ABABAB
```

Sample Output

```
Yes
```

Sample Input

```
ABC
```

Sample Output

```
No
```

Sample Input

```
XYXYX
```

Sample Output

```
Yes
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. **Mobile number validation**

Let's implement the logic for mobile number validation using StringBuilder and embed it in our program. Mobile number should precede with "+91", followed by 10 digits. The indexOf() method returns index of given character value or substring. If it is not found, it returns -1. Write a program to validate the mobile number given as input. Use indexOf() to check whether "+91" is present or not.
Create a driver class called Main. In the Main method, obtain the inputs from the console, validate the mobile number and prompt the user as given in sample I/O.

Input Format

Input consist of a mobile number

Output Format

Output prints whether the mobile number is valid or not

Sample Input

```
+919874653210
```

Sample Output

```
Mobile number valid
```

Sample Input

```
9874653210
```

Sample Output

```
Mobile number invalid
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. **Camel case**

Camel case (stylized as camelCase or CamelCase) is the practice of writing compound words or phrases such that each word or abbreviation in the middle of the phrase begins with a capital letter, with no intervening spaces or punctuation. Event names should be entered in camel case format. But many users failed to follow this convention. To maintain uniformity, you have to change all the event names into camel case. Write a program to convert event names to camel case format.
Create a driver class called Main. In the Main method, obtain the inputs from the console and print the names of the events in camel case.

Input Format

Input consist of the event name

Output Format

Output prints every first letter of the word in uppercase

Sample Input

```
book sale
```

Sample Output

```
BookSale
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **Balls for Challenge**

The Circoloco Children Carnival is the City's largest and successful event dedicated to children and families. The main focus at the carnival is the workshop arena where kids can participate in engaging and educational activities.
Charlie, a little boy accompanied by his Mom visited the fair, where he participated at the "Balls for Challenge" activity. He was given many balls of white and black colors. During the play, he arranged the balls into two rows both consisting of N number of balls. These two rows of balls are given to you in the form of strings X, Y. Both these string consist of 'W' and 'B', where 'W' denotes a white colored ball and 'B' a black colored.
Other than these two rows of balls, Charlie has an infinite supply of extra balls of each color. He wants to create another row of N balls, Z in such a way that the sum of hamming distance between X and Z, and hamming distance between Y and Z is maximized.
Hamming Distance between two strings X and Y is defined as the number of positions where the color of balls in row X differs from the row Y ball at that position. e.g. hamming distance between "WBB", "BWB" is 2, as at position 1 and 2, corresponding colors in the two strings differ. As there can be multiple such arrangements of row Z, Charlie wants you to find the lexicographically smallest arrangement which will maximize the above value.

Input Format

First line of the input will contain a string X denoting the arrangement of balls in first row.
Second line of the input will contain the string Y denoting the arrangement of balls in second row.

Output Format

Output a single line containing the string of length N denoting the arrangement of colors of the balls belonging to row Z.
Refer sample input and output for formatting specifications.

Sample Input

```
WBWB
WBBB
```

Sample Output

```
BWWW
```

Sample Input

```
BBBW
BWBB
```

Sample Output

```
WBWB
```

Sample Input

```
WWWW
BBBB
```

Sample Output

```
WBWB
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. **Casper at the Carnival**

The Circoloco Children Carnival is the City's largest and successful event dedicated to children and families. Casper is a smart little boy who loves eating cookies and drinking fresh juices. He visits the carnival with his parents and is going to spend N minutes at the event ground. Each minute he either eats a cookie or drinks fresh juice. Cookies are very sweet and thus Casper's parents have instructed him to drink fresh juice in the next minute, after eating a cookie.
You are given whether he ate a cookie or drank fresh juice in each of the N minutes. Your task is to check if Casper followed his parents' instructions. That is, you need to verify whether after each eaten cookie he drinks fresh juice in the next minute.

Input Format

The first line of the input contains an integer N denoting the number of minutes.
The second line of the input contains N space-separated strings S1, S2, ...,SN. The string Si is either "cookie" (if Casper eats a cookie in the i-th minute) or "juice" (otherwise).

Output Format

Output a single line containing the answer — "Yes"(without quotes) if Casper followed his parents' instructions, and "No"(without quotes) otherwise, both without the quotes.

Refer sample input and output for formatting specifications.

Sample Input

```
5
cookie cookie juice juice juice
```

Sample Output

```
No
```

Sample Input

```
7
cookie juice juice cookie juice cookie juice
```

Sample Output

```
Yes
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7. **Adjacent characters**

Given a string, write a program to compute a new string where identical chars that are adjacent in the original string are separated from each other by a "*"

Input Format

Input consists of a string.

Output Format

Output prints the newly formed string.

Sample Input

```
hello
```

Sample Output

```
hel*lo
```

Sample Input

```
aaabbb
```

Sample Output

```
a*a*ab*b*b
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8. **Caption Contest**

Exeter Caption Contest is a competition open to all writers worldwide. The entrants will have one day to compose and submit a caption that will be based on the theme posted on the competition page.
Robin, a creative writer had penned two captions for the contest but he unknowingly misplaced them. After searching long, he managed to locate his captions, but some letters in them have become unreadable. The captions were in two very old sheets of paper, each of which originally contained a string of lowercase English letters. The strings on both the sheets have equal lengths. Robin would like to estimate the difference between these strings. Let's assume that the first string is named S1, and the second S2. The unreadable symbols are specified with the question mark symbol '?'. The difference between the strings equals to the number of positions i, such that S1i is not equal to S2i, where S1i and S2i denote the symbol at the i th position in S1 and S2, respectively.
Robin would like to know the minimal and the maximal difference between the two strings, if he changes all unreadable symbols to lowercase English letters. Robin is not an expertise in programming and so he needs your help solving this problem!

Input Format

The first line of the input contains a string S1.
The second line of the input contains a string S2.
Both strings consist of lowercase English letters and question marks in places where the symbols are unreadable.

Output Format

Output the minimal and the maximal difference between two given strings separated with a single space.
Refer sample input and output for formatting specifications.

Sample Input

```
a?c
??b
```

Sample Output

```
1 3
```

Sample Input

```
???a
???a
```

Sample Output

```
0 3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9. Consider a fleet of soldiers of a country are being assembled for a rehearsal session, the enemy country secretly surrounded them and has a special strategy to kill the soldiers in a particular pattern.

Assume that the soldiers are standing in a single straight line. The enemies will repeatedly scan through this line and kill soldiers who are all matching the given pattern.

Find the list of soldiers who are surviving at last or find if all of them are killed.

Here soldiers are represented as alpha-numeric letters, irrespective of cases.

Implement "**FindRemainingSoldiers**" class with "**defeatSoldiers(String soldiers, String pattern)**" method to find the left out soldiers if any, else print "**Defeat**" as result.

Example:
soldiers: xAbcyAAbcbAbccz
pattern: Abc
Iteration:
0: x**Abc**yAAbcbAbccz
1: xyA**Abc**bAbccz
2: xyAb**Abc**cz
3: xy**Abc**z
4: xyz
Output: xyz

Input Format

Input consists of two lines.
First line represent "**fleet of soldiers**" (alpha-numeric string)
Second line represent the kill "**pattern**" (alpha-numeric string)

Output Format

Output should be the left out soldiers string if any, else "**Defeat**"

Constraints

Fleet of soldiers: **{ 0-9, a-z, A-Z }** (1 <= soldiers <= 100)
Pattern: **{ 0-9, a-z, A-Z }** (1 <= pattern <= 100)

Sample Input

```
AbAAbcbcc
Abc
```

Sample Output

```
Defeat
```

Sample Input

```
AAbAbccc
Abc
```

Sample Output

```
Ac
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10. Write a program to convert a String to an int.
Note: If the string contains character then print 0.
Eg. a1b2 here the input contains character, so conversion is not possible.

Input Format

A number as string

Output Format

Integer converted from string

Sample Input

```
234
```

Sample Output

```
234
```

Sample Input

Sample Output

a12	0

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
One fine morning, a minister from Emperor Akbar's
gathered
```

Output

```
String is found in the sentence
```

Weightage - 20

Input

```
Child can study at Hogwarts and learn about life
Kids
```

Output

```
String is not found in the sentence
```

Weightage - 20

Input

```
Each story helps a child enter a completely diffe
 children
```

Output

```
String is not found in the sentence
```

Weightage - 20

Input

```
Our thousands of readers have appreciated our cont
thousands
```

Output

```
String is found in the sentence
```

Weightage - 20

Input

```
Our thousands of readers have appreciated our cont
 efforts
```

Output

```
String is found in the sentence
```

Weightage - 20

Sample Input

```
One fine morning, a minister from Emperor Akbar's
stolen
```

Sample Output

```
String is found in the sentence
```

Sample Input

```
One fine morning, a minister from Emperor Akbar's
 stolen
```

Sample Output

```
String is found in the sentence
```

Sample Input

Sample Output

One fine morning, a minister from Emperor Akbar's account	String is not found in the sentence

Sample Input **Sample Output**

One fine morning, a minister from Emperor Akbar's account	String is not found in the sentence

Solution

```
import java.io.*;
import java.util.*;
 class Main {
public static void main(String [] args) {
    String sentence,search;
    Scanner sc = new Scanner(System.in);
    sentence = sc.nextLine();
    search = sc.nextLine();
    if(sentence.contains(search.trim())) {
        System.out.println("String is found in the sentence");
    }
    else {
        System.out.println("String is not found in the sentence");
    }
}
}
```

Q2

Test Case

Input **Output**

ABABAB	Yes

Weightage - 10

Input **Output**

ABC	No

Weightage - 10

Input **Output**

XYXYX	Yes

Weightage - 10

Input **Output**

IJIJIJ	Yes

Weightage - 15

Input

KLKLK

Output

Yes

Weightage - 15

Input

IJK

Output

No

Weightage - 10

Input

IJIJK

Output

No

Weightage - 20

Input

WXYZ

Output

No

Weightage - 10

Sample Input

ABABAB

Sample Output

Yes

Sample Input

ABC

Sample Output

No

Sample Input

XYXYX

Sample Output

Yes

Solution

```
import java.io.*;
import java.util.*;
class AlternatingCode {
    public static void main(String [] args) {
```

```
int i,s=0;
String a;
char x,y;
Scanner sc = new Scanner(System.in);
a = sc.next();
x = a.charAt(0);
y = a.charAt(1);
if(x==y) {
    s=1;
    }
else{
for(i=0;i<a.length();i++)
{
    if(i%2==0) {
        if(x!=a.charAt(i)) {
            s=1;
            break;
            }
        }
    else if(y!= a.charAt(i)) {
        s=1;
        break;
            }
        }
        }
        if(s==1) {
            System.out.println("No");
        }
        else {
            System.out.println("Yes");
        }

    }
}
```

Q3

Test Case

Input

```
+919874653210
```

Output

```
Mobile number valid
```

Weightage - 20

Input

```
9874653210
```

Output

```
Mobile number invalid
```

Weightage - 20

Input

```
+919633319072
```

Output

```
Mobile number valid
```

Weightage - 20

Input

```
9633319072
```

Output

```
Mobile number invalid
```

Weightage - 20

Input

```
+91874444883
```

Output

```
Mobile number invalid
```

Weightage - 20

Sample Input

```
+919874653210
```

Sample Output

```
Mobile number valid
```

Sample Input

```
9874653210
```

Sample Output

```
Mobile number invalid
```

Solution

```
import java.io.*;
import java.util.*;
class Main {
public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    String mobile = sc.nextLine();
    if(mobile.substring(0, 3).compareTo("+91") == 0 && mobile.substring(3).length() ==  10) {
        System.out.println("Mobile number valid");
    }
    else {
        System.out.println("Mobile number invalid");
     }
}
}
```

Q4

Test Case

Input

```
book sale
```

Output

```
BookSale
```

Weightage - 20

Input

```
food fest
```

Output

```
FoodFest
```

Weightage - 20

Input

```
fish stall
```

Output

```
FishStall
```

Weightage - 20

Input

```
sports meet under18
```

Output

```
SportsMeetUnder18
```

Weightage - 20

Input

```
cultural fest for upcoming talents
```

Output

```
CulturalFestForUpcomingTalents
```

Weightage - 20

Sample Input

```
book sale
```

Sample Output

```
BookSale
```

Solution

```
import java.io.*;
import java.util.*;
class Main {
public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    char res [] = str.toCharArray();
    int index = 1,i;
    res[0] = Character.toUpperCase(res[0]);
    for( i=1;i<res.length;i++) {
        if(res[i] == ' ') {
            res[i+1] = Character.toUpperCase(res[i+1]);
            continue;
        }
        else {
            res[index++] = res[i];
        }
    }
    for(i=0;i<index;i++) {
        System.out.print(res[i]);
    }
}
}
```

Q5

Test Case

Input

```
WBWB
WBBB
```

Output

```
BWWW
```

Weightage - 10

Input

```
BBBW
BWBB
```

Output

```
WBWB
```

Weightage - 10

Input

```
WWWW
BBBB
```

Output

```
WBWB
```

Weightage - 10

Input

```
WWBBWB
WBBBBW
```

Output

```
BWWWBB
```

Weightage - 15

Input

```
WBBBWW
BBBWWW
```

Output

```
WWWWBB
```

Weightage - 15

Input

```
WWBBWBWB
BBWWBWBW
```

Output

```
WBBWWWWW
```

Weightage - 20

Input

```
BBWWBBWW
WWBBWWBB
```

Output

```
BWWBBWWB
```

Weightage - 20

Sample Input

```
WBWB
WBBB
```

Sample Output

```
BWWW
```

Sample Input

```
BBBW
BWBB
```

Sample Output

```
WBWB
```

Sample Input

```
WWWW
BBBB
```

Sample Output

```
WBWB
```

Solution

```
import java.io.*;
import java.util.*;
class BallsChallenge {
    public static void main(String [] args) {
        int i,c=0;
        String str1,str2;
        Scanner sc = new Scanner(System.in);
        str1 = sc.next();
        str2 = sc.next();
        for(i=0;i<str1.length();i++)
        {
            if(str1.charAt(i)==str2.charAt(i))
            {
                if(str1.charAt(i) == 87) {
                    System.out.print("B");
                }
                else {
                    System.out.print("W");
                }
            }
            else
            {
                c++;
                if(c%2 == 0) {
                    System.out.print(str2.charAt(i));
                }
                else {
                    System.out.print(str1.charAt(i));
                }
            }
        }

    }
}
```

Q6

Test Case

Input

Output

5 cookie cookie juice juice juice	No

Weightage - 10

Input **Output**

7 cookie juice juice cookie juice cookie juice	Yes

Weightage - 10

Input **Output**

5 cookie juice cookie juice cookie	No

Weightage - 10

Input **Output**

8 cookie juice juice cookie juice juice cookie juice	Yes

Weightage - 15

Input **Output**

8 cookie juice juice cookie juice juice cookie cooki	No

Weightage - 15

Input **Output**

10 cookie juice juice cookie juice juice cookie juice	Yes

Weightage - 20

Input **Output**

10 cookie juice juice cookie juice juice cookie juice	No

Weightage - 20

Sample Input **Sample Output**

5 cookie cookie juice juice juice	No

Sample Input

```
7
cookie juice juice cookie juice cookie juice
```

Sample Output

```
Yes
```

Solution

```
import java.io.*;
import java.util.*;
class CasperCarnival {
    public static void main(String [] args) {
        int i,s,n;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        String str[] = new String[n];
        if(n>1) {
            s=0;
        }
        else {
            s=1;
        }
        for(i=0;i<n;i++)
        {
            str[i] = sc.next();
            if(i>0)
            {
                if(str[i-1].equals("cookie") && !(str[i].equals("juice"))) {
                    s=1;
                    }
            }
            if(i==n-1) {
                if(str[n-1].equals("cookie")) {
                    s=1;
                    }
                }
        }
        if(s==1) {
            System.out.println("No");
        }
        else {
            System.out.println("Yes");
            }
        }
    }
```

Q7

Test Case

Input

```
hello
```

Output

```
hel*lo
```

Weightage - 10

Input

```
aabb
```

Output

```
a*ab*b
```

Weightage - 10

Input	Output
abba	ab*ba

Weightage - 10

Input	Output
abbaab	ab*ba*ab

Weightage - 15

Input	Output
aaabbb	a*a*ab*b*b

Weightage - 15

Input	Output
abbabaab	ab*baba*ab

Weightage - 20

Input	Output
aaaabbbb	a*a*a*ab*b*b*b

Weightage - 20

Sample Input	Sample Output
hello	hel*lo

Sample Input	Sample Output
aaabbb	a*a*ab*b*b

Solution

```
import java.io.*;
import java.util.*;
class AdjacentCharacters {
    public static void main(String [] args) {
        String s;
        Scanner sc = new Scanner(System.in);
        s = sc.next();
        pairStar(s,0);
        System.out.println(s1);
        for(int i=0; i<s.length();i++) {
            s1 = s1+s.charAt(i);
            if(i == s.length()-1) {
                return;
            }
            if(s.charAt(i) == s.charAt(i+1)) {
                s1 = s1+'*';
            }
        }
    }
    static String s1="";
    private static void pairStar(String s, int i) {
        s1 = s1+s.charAt(i);
        if(i == s.length()-1) {
            return;
        }
        if(s.charAt(i) == s.charAt(i+1)) {
            s1 = s1+'*';
        }
        pairStar(s,i+1);
    }
}
```

Q8

Test Case

Input

```
a?c
??b
```

Output

```
1 3
```

Weightage - 10

Input

```
???a
???a
```

Output

```
0 3
```

Weightage - 10

Input

```
a??b
???a
```

Output

```
1 4
```

Weightage - 10

Input

Output

abb??ab baa??bb	4 6

Weightage - 15

Input	Output
abb?aba bab??aa	3 5

Weightage - 15

Input	Output
ab???aabbb a???????aa	2 9

Weightage - 20

Input	Output
ababab??bb baba????aa	6 10

Weightage - 20

Sample Input	Sample Output
a?c ??b	1 3

Sample Input	Sample Output
???a ???a	0 3

Solution

```
import java.io.*;
import java.util.*;
class CaptionContest {
    public static void main(String [] args) {
        int i,min=0,max=0;
        String str1,str2;
        Scanner sc = new Scanner(System.in);
        str1 = sc.next();
        str2 = sc.next();
        for(i=0;i<str1.length();i++)
        {
            if(str1.charAt(i)!=str2.charAt(i)) {
                if((str1.charAt(i)!=63)&&(str2.charAt(i)!=63)) {
                    min++;
                    max++;
                    }
```

```
                else {
                    max++;
                    }
            }
            else if((str1.charAt(i)==str2.charAt(i))&&(str1.charAt(i)==63)) {
                max++;
                }
        }
        System.out.print(min+" "+max);
    }
}
```

Q9

Test Case

Input

```
xC00xC0xC00707xC0077
xC007
```

Output

```
Defeat
```

Weightage - 10

Input

```
b45b45Ab4b45b45AntAnt5AntntqAnt
b45Ant
```

Output

```
b45qAnt
```

Weightage - 10

Input

```
ioioXC90TXioXC90TCioXC90T9ioXC90T0ioXC90TT
ioXC90T
```

Output

```
Defeat
```

Weightage - 10

Input

```
LmaZdaAmaZdaNmaZdaCmaZdaEmaZdaRmaZda
maZda
```

Output

```
LANCER
```

Weightage - 10

Input

```
1
xYz
```

Output

```
1
```

Weightage - 20

Input

```
SkSkYlIneSkYlIneYlISkYSkYlInelIneSkYlIneneSkYlSkYlIn
Rx7
```

Output

```
SkSkYlIneSkYlIneYlISkYSkYlInelIneSkYlIneneSkYlSkYlIn
```

Weightage - 20

Input

```
SSiaNiSSSiaNiSiaNaSiaNNiaNSiaNaSSiaNiaNN
SiaN
```

Output

```
Defeat
```

Weightage - 20

Sample Input

```
AbAAbcbcc
Abc
```

Sample Output

```
Defeat
```

Sample Input

```
AAbAbccc
Abc
```

Sample Output

```
Ac
```

Solution

Header

```
import java.util.*;

class FindLeftSoldiers {



public static String defeatSoldiers(String soldiers, String pattern) {
    while (soldiers.length() > 0) {

        int idx = soldiers.indexOf(pattern);
        if (idx == -1) {
            break;
        }

        soldiers = soldiers.replaceFirst(pattern, "");

    }

    return soldiers;
}
```

Footer

```
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        String soldiers = in.next();
        String pattern = in.next();

        in.close();

        String result = defeatSoldiers(soldiers, pattern);

        if (result.length() == 0) {
```

```
            System.out.println("Defeat");
        } else {
            System.out.println(result);
        }

    }

}
```

Q10

Test Case

Input

```
786
```

Output

```
786
```

Weightage - 25

Input

```
456
```

Output

```
456
```

Weightage - 25

Input

```
4851
```

Output

```
4851
```

Weightage - 25

Input

```
a1234b
```

Output

```
0
```

Weightage - 25

Sample Input

```
234
```

Sample Output

```
234
```

Sample Input

```
a12
```

Sample Output

```
0
```

Solution

```
import java.util.*;
import java.lang.*;
```

```
import java.io.*;

class Q01Medium_StringAdv {

    public static int convert(String str) {
        int val = 0;
        try {
            val = Integer.parseInt(str);
        } catch (NumberFormatException e) {
            // Invalid String
        }
        return val;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System. in);
        String str = input.nextLine();

        int val = convert(str);
        System.out.println(val);
    }
}
```