


```

16 v         else if(space == ones && flag == 0){
17             temp.append(Character.toString(g.charAt(i)));
18         }
19 v         else if(space == ones && flag == 1){
20             temp1.append(Character.toString(g.charAt(i)));
21         }
22     }
23     space = 0 ;
24     flag = 1;
25     n = n /10;
26 }
27 rew m = new rew();
28 System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));
29 }
30 }
31 v class rew{
32 v     String r(String a){
33         int le = a.length(),n,q;
34         StringBuffer temp3 = new StringBuffer();
35 v         if(le % 2 == 1){
36             n = ((int)(le/2));
37             q = ((int)(le/2));
38         }
39 v         else{
40             n = ((int)(le/2)) - 1;
41             q = ((int)(le/2));
42         }
43 v         for(int i = n;i >= 0;i--){
44             temp3.append(Character.toString(a.charAt(i)));
45         }
46 v         for(int i = q;i < le;i++){
47             temp3.append(Character.toString(a.charAt(i)));
48         }
49         return temp3.toString();
50     }
51 }
52 }

```

Input	Expected	Got
Today is a Nice Day 41	iNce doTday	iNce doTday
Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes

Passed all tests!

Question 2

Correct

Marked out of 5.00

[Flag question](#)

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases.

Note:

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

input1 = ww:ii:pp:rr:oo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

input1 = zx:za:ee

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24
max – min will be 26 – 24 = 2
Alphabet which comes in 2nd position is b
Word2 is za, both are not same alphabets
position value of z is 26
position value of a is 1
max – min will be 26 – 1 = 25
Alphabet which comes in 25th position is y
word3 is ee, both are same hence take e
Hence the output is BYE

For example:

Input	Result
ww:ii:pp:rr:oo	WIPRO
zx:za:ee	BYE

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 class diff{
3     char different(char a, char b){
4         if ((int)a != (int)b)
5             return (char)((int)'a' + ((int)a-(int)b) - 1);
6         return a;
7     }
8 }
9 public class Main{
10     public static void main(String[] args){
11         Scanner scan = new Scanner(System.in);
12         diff z = new diff();
13         String q = scan.nextLine();
14         StringBuffer ans = new StringBuffer();
15         StringBuffer temp = new StringBuffer();
16         for(int i = 0; i < q.length(); i++){
17             if(q.charAt(i) == ':'){
18                 temp.append(" ");
19             }
20             else{
21                 temp.append(Character.toString(q.charAt(i)));
22             }
23         }
24         String h = temp.toString();
25         for(int i = 0; i < temp.length(); i++){
26             if(i%3 == 0){
27                 ans.append(Character.toString(z.different(h.charAt(i), h.charAt(i+1))));
28             }
29         }
30         System.out.print(ans.toString().toUpperCase());
31     }
32 }
33

```

Input	Expected	Got
ww:ii:pp:rr:oo	WIPRO	WIPRO
zx:za:ee	BYE	BYE

Passed all tests!

Question **3**

Correct

Marked out of 5.00

[Flag question](#)

Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

Assumption 1:

There will either be alphabets, white spaces or null in both the inputs.

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

For example:

Test	Input	Result
1	apple orange	rponlgea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2
3
4 public class HelloWorld {
5     public static void main(String[] args) {
6         Scanner scan = new Scanner(System.in);
7         String a = scan.nextLine();
8         String b = scan.nextLine();
9         StringBuffer ab = new StringBuffer();
10        if(a.trim().isEmpty() && b.trim().isEmpty()){
11            System.out.print("null");
12        }
13        else{
14            for(int i = 0; i < a.length(); i++){
15                if (a.charAt(i) != ' ') {
16                    ab.append(Character.toString(a.charAt(i)));
17                }
18            }
19            for(int i = 0; i < b.length(); i++){
20                if (b.charAt(i) != ' '){
21                    ab.append(Character.toString(b.charAt(i)));
22                }
23            }
24            char[] d = ab.toString().toCharArray();
25            Arrays.sort(d);
26            for(int i = d.length - 1; i >= 1; i--){
27                if(d[i] != d[i-1])
28                    System.out.print(d[i]);
29            }
30            System.out.print(d[0]);
31        }
32    }
33 }
34 }
35 }
```

Test	Input	Expected	Got
1	apple orange	rponlgea	rponlgea
2	fruits are good	utsroigfeda	utsroigfeda
3		null	null

Passed all tests!

[Finish review](#)

[◀ Lab-06-MCQ](#)

Jump to...

[Return second word in Uppercase ▶](#)