

Talent Battle 100 Days Coding Series

Akhil has many balls of white and black colors. One day, he was playing with them. 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, Akhil 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**, Akhil wants you to find the lexicographically smallest arrangement which will maximize the above value.

Input

- The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows:
- First line of each test case will contain a string **X** denoting the arrangement of balls in first row
- Second line will contain the string **Y** denoting the arrangement of balls in second row.

Output

- For each test case, output a single line containing the string of length **N** denoting the arrangement of colors of the balls belonging to row **Z**.

Constraints

- $1 \leq T \leq 3$

Sample Input

1

WBWB

WBBB

Sample Output

BWBW

Talent Battle 100 Days Coding Series

C++

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int T;
```

```
    string S1,S2;
```

```
    cin>>T;
```

```
    getline(cin,S1);
```

```
    while(T--){
```

```
        getline(cin,S1);
```

```
        getline(cin,S2);
```

```
        string Z="";
```

```
        for(int i=0;i<S1.size();i++){
```

```
            if(S1[i]==S2[i]){
```

```
                if(S1[i]=='B'){Z+='W';}
```

```
            else{Z+='B';}
```

```
        }
```

```
    else{
```

```
        Z=Z+'B';
```

```
    }
```

```
    }
```

```
    cout<<Z<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

TalentBattle

Talent Battle 100 Days Coding Series

Java

```
import java.io.*;

class Main
{
    public static void main(String[] ar) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter out = new PrintWriter(System.out);
        int t = Integer.parseInt(br.readLine());
        while(t-->0)
        {
            String s1 = br.readLine();
            String s2 = br.readLine();
            int len=s1.length();
            String str="";
            for(int i=0;i<len;i++)
            {
                if(s2.charAt(i)==s1.charAt(i))
                {
                    if(s1.charAt(i)=='W')
                        out.print('B');
                    else
                        out.print('W');
                }
                else
                    out.print('B');
            }
            out.println();
            out.flush();
        }
    }
}
```

```
    }  
    }  
}
```

Python

```
for _ in range(int(input())):
```

```
    x=input()
```

```
    y=input()
```

```
    s=""
```

```
    for i in range(len(x)):
```

```
        if x[i]==y[i] and x[i]=='B':
```

```
            s+='W'
```

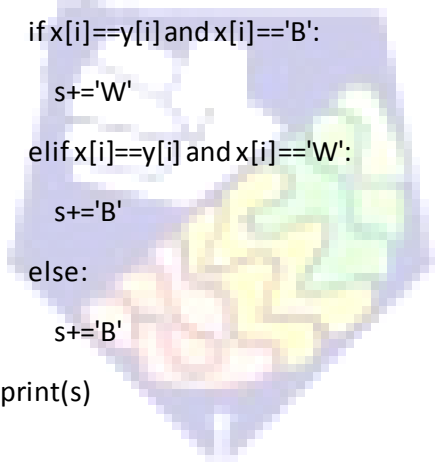
```
        elif x[i]==y[i] and x[i]=='W':
```

```
            s+='B'
```

```
        else:
```

```
            s+='B'
```

```
    print(s)
```



TalentBattle