Assignment Case	BINUS UNIVERSITY Software Laboratory Center
COMP6047 Algorithm and Programming	
Computer Science	**<Case Code>**
Valid on *Compact Semester Year 2019/2020*	**Revision 00**

Soal
Case

xorxorxor

Billy loves to encrypt and decrypt the code he found. One day, Billy found a unique encrypted text in book he currently read. He found out how to decrypt the text but find it too much to decrypt by himself. Therefore, he asked you to help him decrypt the text by using a program. He sends you a file named "**testdata.in**" and give you the instruction.

The first to do is you have to manipulate each character's ASCII Code in the text using bitwise operator **xor** (^) against **8**. The result of the manipulation is a string in format "$c_1d_1c_2d_2c_3d_3\dots$", where c_i is a character and d_i is an integer denoting how many concurrent c_i before the next character. You have to convert the integer into d_i character c_i to decrypt it.

Format Input

The file consists of multiple test cases separated by line break. Each test case consists of a text S, the encrypted text.

Format Output

For each test case, print "Case #X: " where X representing the test case number. Following this is the decrypted text. If the decryption process failed, print "Corrupted format" instead.

Constraints

$2 \le |S| \le 50{,}000$

$0 \le d_i \le 9$

<table>
<tr><th>Sample Input (testdata.in)</th><th>Sample Output</th></tr>
<tr><td>`9m9d:g9
``mm
D9g9z9m9e9(9a9x9{9}9e9
`9i9x:q9
m9p9k9m9d:m9f9|9</td><td>Case #1: hello
Case #2: Corrupted format
Case #3: Lorem ipsum
Case #4: happy
Case #5: excellent</td></tr>
</table>

Explanation:

For the first test case, the input is “`9m9d:g9”.

The first step is manipulating the text’s using xor 8, which resulting in “h1e1l2o1”. Each numeric value will determine how many times the character before it show up. Therefore, the output is “hello”.

For the second test case, after xor each character’s ASCII Code using xor 8, the result is “hhee”, where the second and fourth character aren’t integer. Since it can’t be decrypted any further, output “Corrupted format”.

Note:

Don't forget to add the newline character after printing the output.