UVa Online Judge

909 The BitPack Data Compression Problem

Current applications must deal with every sort of data. Some data sources are huge and some of these must be manipulated as a whole (an image for example). This context gave birth to a new area of study called data compression.

A data compression operation has two opposite faces: the compression and the decompression. Normally, when a standard is published by ISO or CCITT it only describes the simplest operation leaving the other to the application programmers.

The CCITT text for the BitPack algorithm describes the data decompression as in the following:

```
While (there is input)
    n <-- read next byte
    If (0<= n <= 127)
        copy the next n+1 bytes to the output as they are
    Else If (129<= n <=255)
        copy the next byte n-128+1 times to the output
    Else If n=128
        do nothing
```

Your task consists of writing a program that performs the opposite operation: the data compression, in such a way that a file of minimal size is obtained.

Input

The input file contains several test cases, each of them is a simple byte sequence on a line by itself.

Output

For each test case, the output will consist of a line with compressed content.

From this line and applying the algorithm described above it should be possible to retrieve the original case.

Note: in the examples below [*ddd*] stands for a character with ASCII code equal to *ddd*.

Sample Input

```
aaaaaaaarstqahbbbbbbb
aaaaaaaaaa
abcdefghij
```

Sample Output

```
[135]a[5]rstqah[134]b
[137]a
[9]abcdefghij
```