ACM Qualifier 2018: Run-length Encoding, Run!

Forrest lives in a prehistoric era of “dial-up Internet.” Unlike the fast streaming of today’s broadband era, dial-up connections are only capable of transmitting small amounts of text data at reasonable speeds. Forrest has noticed that his communications typically include repeated characters, and has designed a simple compression scheme based on repeated information. Text data is encoded for transmission, possibly resulting in a much shorter data string, and decoded after transmission to reveal the original data.

The compression scheme is rather simple. When encoding a text string, repeated consecutive characters are replaced by a single instance of that character and the number of occurrences of that character (the character’s run length). Decoding the encoded string results in the original string by repeating each character the number of times encoded by the run length. Forrest calls this encoding scheme run-length encoding. (We don’t think he was actually the first person to invent it, but we haven’t mentioned that to him.)

For example, the string `HHHeelllo` is encoded as `H3e2l3o1`. Decoding `H3e2l3o1` results in the original string. Forrest has hired you to write an implementation for his run-length encoding algorithm.

Input Format

Input consists of a single line of text. The line starts with a single letter: `E` for encode or `D` for decode. This letter is followed by a single space and then a message. The message consists of 1 to 100 characters.

Each string to encode contains only upper- and lowercase English letters, underscores, periods, and exclamation points. No consecutive sequence of characters exceeds 9 repetitions.

Each string to decode has even length. Its characters alternate between the same characters as strings to encode and a single digit between 1 and 9, indicating the run length for the preceding character.

Output Format

On an input of `E` output the run-length encoding of the provided message. On an input of `D` output the original string corresponding to the given run-length encoding.

Sample Input 0

```
E HHHeellloWooorrrrlld!!
```

Sample Output 0

```
H3e2l3o1W1o3r4l2d1!2
```

Sample Input 1

```
D H3e2l3o1W1o3r4l2d1!2
```

Sample Output 1

```
HHHeellloWooorrrrlld!!
```