ASDAN 阿思丹 China | Title

Welcome   Prepare   Competitions

Hide menu

2024 ACSL 全明星正式考试 (Junior)

Run-Length Encoding
ACSL Word Search

End competition

Run-Length Encoding

Time Remaining   2:58   hrs:mins

CLASS / SOURCE NAME

Nameyourclass acslrun.

PROBLEM: *Run-length encoding* is a data compression technique that works well on data in which values repeat frequently, such as black-and-white images. The idea is to replace runs of the same character by that character followed by a count of how many times that character appears. This compression technique is *lossless*, meaning that the original source can be reconstructed from the compressed version.

For example, consider the input string ABBCEEEEAAAAADDDAAA. Every count will be a hexadecimal number, one less than the number of occurrences of that character. The encoded version would be A0B1C0E3A5D2A2. The original string was 20 characters; the encoded string is only 14 characters. If there are more than 16 occurrences of a character, output the character multiple times. For example, a string of 5 Cs followed by 30 Bs followed by 2 As would be compressed to C4BFBDA1.

In the **encoding** part of this program, you will be given an alphanumeric string and your program will return the encoded version. In the **decoding** part of this program, you will be given a string which is a run-length encoding of an alphanumeric string. You need to decode it and return it as a single string. The data will start with an **E** for encoding and a **D** for decoding, followed by a string as described above.

Your program must also support the following variation, indicated by an **EV** or a **DV**: rather than breaking the repeated character into chunks of 16, simply output the number of times, in hexadecimal, that the repeated character appears. When the count is 2 or more hexadecimal digits, it is preceded and followed by a dash. In this variation, the count is the number of occurrences, not one less. For example, ABBCEEEEAAAAADDDAAA is encoded as A1B2C1E4A6D3A3 and a string of 5 Cs followed by 30 Bs followed by 2 As would be compressed to C5B-1E-A2.

INPUT: There will be two strings of data. The first one or two character string will represent the code (E, D, EV, DV). The second string will be the message to be encoded or decoded. All messages will have a valid format. All hex digits in encoded strings will be capitalized. Any character on the keyboard can be used in the message except for a "-", which is only used as a delimiter.

OUTPUT: Output a string that represents either the encoded message or the decoded message depending on the code that is input.

SAMPLE INPUT The 2nd string in each data set is a single string; for readability, it appears over multiple lines.	SAMPLE OUTPUT The output is a single string; for readability, it appears over multiple lines.
E ABBCEEEEAAAAAADDDAAACCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAA	A0B1C0E3A5D2A2C4BFBDA1
EV ABBCEEEEAAAAAADDDAAACCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAA	A1B2C1E4A6D3A3C5B-1E-A2
D a0b1a2dAfFfFfEE2	abbaaadddddddddddfffffffffffffffffffffffffffffffffffffffffffffffffEEE
DV a1b2a3dBf-2F-E3	abbaaadddddddddddfffffffffffffffffffffffffffffffffffffffffffffffffEEE

--中文仅供参考，以英文为准--

问题: *行程长度编码*是一种数据压缩技术，适用于存在大量重复值的数据（如黑白图像）。其原理是将多个连续重复的字符替换为该字符后紧跟其重复出现的次数。这种压缩技术是*无损的*，这意味着可以用压缩版本重建原始数据源。

例如，输入字符串 ABBCEEEEAAAAAADDDAAA。其编码过程中的每次计数都用一个 16 进制数表示，并且每次计数都比字符实际连续重复次数少 1。编码后，该字符串将变为 A0B1C0E3A5D2A2。原始的字符串包含 20 个字符，而编码后的字符串则缩减至 14 个字符。如果某个

字符连续重复出现的次数超过 16 次，则需要多次输出该字符。例如，一个由 5 个 C 后跟 30 个 B 再跟 2 个 A 组成的字符串将被压缩为 C4BFBDA1。

在程序的**编码**部分，你将接收到一个字母数字字符串，随后你所编写的程序会输出其对应的编码形式。在程序的**解码**部分，你将接收到一个已经经过行程长度编码处理的字母数字字符串。你需要将其解码并以单个字符串的形式返回。一个字符串用字母 E 引导，表示该字符串需编码；一个字符串用字母 D 引导，表示该字符串需解码。

你所编写的程序还必须支持以下用 EV 或 DV 引导的变体：无需将连续重复出现的字符以 16 个为一组进行划分，直接输出字符连续重复出现的次数即可，次数用16 进制表示。当字符连续重复出现次数表示为 2 位或更多位的16 进制数字时，这个 16 进制数前后要用破折号隔开。在这种变体中，计数结果即字符实际重复出现的次数，无需减 1。例如 ABBCEEEEAAAAAADDDAAA 被编码为 A1B2C1E4A6D3A3；由 5 个 C，后接 30 个 B，随后是 2 个 A 组成的字符串可能被压缩为 C5B-1E-A2。

输入: 两个数据字符串。第一个字符串由 1 或 2 个字符组成，代表编码类型 (E，D，EV，DV)。第二个字符串是待编码或解码的信息。所有信息都遵循有效的格式规范。在编码后的字符串中，所有的 16 进制数字都将用大写字母表示。除了用作破折号的"-"外，键盘上的所有字符都可以用于构成信息的内容。

输出: 根据输入的代码，输出一个字符串，表示编码或解码后的信息。

样例输入 每个数据集中的第 2 个字符串为单个字符串；为便于阅读，该字符串分多行显示。	样例输出 输出结果为单个字符串；为便于阅读，输出结果分多行显示
E ABBCEEEEAAAAAADDDAAACCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAA	A0B1C0E3A5D2A2C4BFBDA1
EV ABBCEEEEAAAAAADDDAAACCCCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAA	A1B2C1E4A6D3A3C5B-1E-A2
D a0b1a2dAfFfFfEE2	abbaaadddddddddddffffffffffffffffffffffffffffffffffffffffffffEEE

DV a1b2a3dBf-2F-E3	abbaaadddddddddddddffffffffffffffffffffffffffffffffffffffffffff fffffEEE

注意:
(1) 样本数据仅为部分测试数据，测试用例全部通过不代表通过本题。
(2) 你必须通过数据库中所有的测试点才能获得该题满分。
(3) java 语言里面的 class name（类名）需要用本题的 Source file name（即：acslrun）。
(4) 平台判分规则为调取 10 次 input 值，每次单独判分。

Compiler [Python 3/CPython ⌄]

```
1    Enter your program code here
```

```
Enter custom input data here (optional)
```

Run code   Submit code

Previous submissions

Click the filename to download

Date   Filename   Score   Status   Compiler

Asdan Challenges are proud to use The Cuttle Platform to host our challenges.
The Cuttle system can be used for maths and science competitions, entrance exams or employee exams.