实验五 LZW 编码仿真实现

一、 实验目的

1. 理解 LZW 算法的原理；
2. 编写 MATLAB 程序实现 LZW 编码。

二、 实验原理

1. LZW 算法简介

LZW(Lempel-Ziv-Welch)编码又称“串表压缩算法”，是 Welch 将 Lempel 和 Ziv 所提出来的无损压缩技术改进后的压缩方法。该算法通过建立一个字符串表，用较短的码字来表示较长的字符串来实现压缩。下面是 LZW 编码的思路：考虑一段字符串数据‘abcabcabc’，假设原始字典如下表所示，每个字符需要一个字节（byte）的存储空间，则经过字典编码后得到的结果用十进制表示为 123123123，直接存储的空间需要 9 个字节。

字符	十进制编号	二进制编号	存储空间
a	1	0000 0001	1 byte
b	2	0000 0010	1 byte
c	3	0000 0011	1 byte

再次观察上面的字符串，可以发现‘abc’是重复的，如果‘abc’能在字典中用一个字节来表示的话，那么，只需要 3 字节就能够存储上面的字符串。

下表是我们新建的一个字典：

字符	十进制编号	二进制编号	存储空间
abc	1	0000 0001	1 byte

通过上面的字典，对字符串数据‘abcabcabc’经过字典编码后得到的十进制结果为 111，只需要 3 字节就可以存储了。

LZW 算法就是对上述过程的一种改进。

2. LZW 编码基本思想

LZW 编码的基本思想如下，首先我们需要初始化一个字典，里面保存了待编码的字符串中所有第一次出现的单个字符和对应的编号，比如 0-9，a-z，A-Z 这些，通常用十进制数 0-255 对单个字符进行编号，或者也可以使用字符在 ASCII

表中的编号。字典准备好之后，开始读取字符串，很显然对于任何常规字符串，每个单一字符都可以得到一个码字，但是如果我们不做处理的话，那么对 n 个字符进行编码后，就得到了 n 个码字，原始字符串完全没有被压缩。所以，在读取每个字符的时候，同时开始对字典进行扩充，不断的将单一字符拼接成字符串，成为字典中不存在的符号，并将拼接的字符串扩充进字典，这样下次再遇到这样的字符组合，就可以仅使用一个编号表示了，从而对原始数据进行了压缩。

3. LZW 编码流程

LZW 编码的具体步骤如下：

(1) 初始词典包含待编码字符串中所有第一次出现的单个字符及其编号，这里可以从数字 1 开始按顺序向后编号，前缀字符（P）初始化设置为空；

(2) 当前读取字符（C） = 待编码字符串中的下一个字符；

(3) 判断字符串（PC）是否在字典中；

- 如果“是”：

  P = PC; // 将 P 和 C 合并为一个字符串赋值给 P；

- 如果“否”：

  ① 将 P 和 C 合并为一个整体 PC 并将其添加到词典；

  ② 输出 P 在字典中对应的编号，构成编码后的码字流；

  ③ 令 P = C； // 将 C 赋值给 P；

(4) 判断待编码字符串中是否还有字符需要编码；

- 如果“是”：

  返回到步骤(2)；

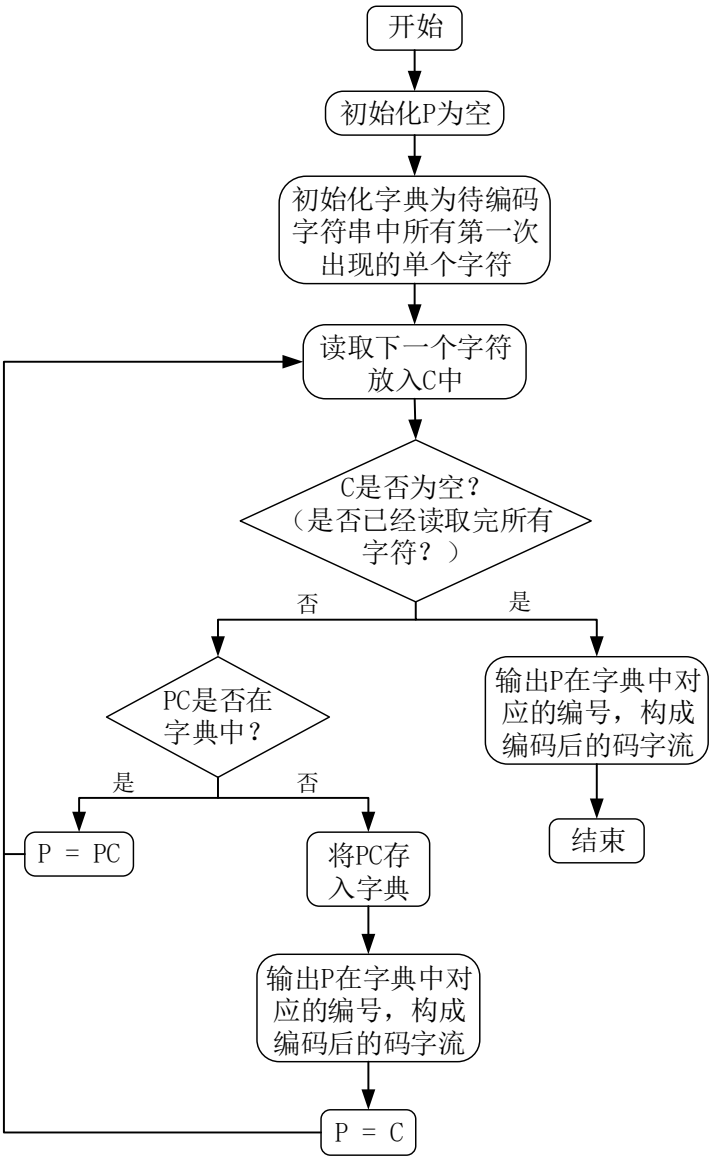
- 如果“否”：

  ① 输出 P 在字典中对应的编号，构成编码后的码字流；

  ② LZW 编码结束。

LZW 编码的流程图如下所示：

其中，P 表示 Previous，用来存储之前的字符，C 表示 Current，用来存储当前读取的字符。



4. LZW 编码示例分析

考虑字符串‘abcbcabcabcd’，按照上述编码步骤演示一遍 LZW 编码过程：

首先，我们需要一个字典，初始字典只保存了字符串中所有第一次出现的单个字符 a-d，编号分别设为 1-4，如下图所示：

字符	编号
a	1
b	2
c	3
d	4

下面是具体的编码过程，即按照实验原理中的 LZW 编码流程图或编码步骤对字符串进行编码。

步数	P	C	PC 在字典中?	输出 P	字典	描述
1	NULL	a	yes			a 在字典中，P=PC
2	a	b	no	1	ab:5	ab 不在字典中，扩充进字典，输出 P 的编号
3	b	c	no	2	bc:6	bc 不在字典中，扩充进字典，输出 P 的编号
4	c	b	no	3	cb:7	cb 不在字典中，扩充进字典，输出 P 的编号
5	b	c	yes			bc 在字典中，P=PC
6	bc	a	no	6	bca:8	bca 不在字典中，扩充进字典，输出 P 的编号
7	a	b	yes			ab 在字典中，P=PC
8	ab	c	no	5	abc:9	abc 不在字典中，扩充进字典，输出 P 的编号
9	c	a	no	3	ca:10	ca 不在字典中，扩充进字典，输出 P 的编号
10	a	b	yes			ab 在字典中，P=PC
11	ab	c	yes			abc 在字典中，P=PC
12	abc	d	no	9	abcd:11	abcd 不在字典中，扩充进字典，输出 P 的编号
13	d	NULL		4		结束，输出 P 的编号

经过上面的演算过程，编码结束后得到的输出码字流为：1 2 3 6 5 3 9 4

字典也被扩充为：

字符	编号
a	1
b	2
c	3
d	4
ab	5
bc	6
cb	7
bca	8
abc	9
ca	10
abcd	11

5. 实验可能用到的 Matlab Function 和数据类型（仅供参考）

input(), strcmp(), find(), strcat(), cell 型数据类型等。

三、 实验预习

回答以下问题：

1. 假设有一字符串为“abcabcabc”，参考实验内容中的 LZW 编码示例分析部分，回答以下问题。

(1) 写出初始字典。
(2) 对上述字符串进行 LZW 编码，仿照实验内容中的 LZW 编码示例分析部分，画出表格。
(3) 写出经过 LZW 编码后得到的码字流和被扩充的字典。

四、 实验内容（要求给出结果截图，源代码放在.m 文件中）

1. 按照实验原理中的 LZW 编码流程，用 MATLAB 软件编写程序实现对任意给定的字符串进行 LZW 编码。要求最终在控制台依次输出初始字典，字符串经过 LZW 编码后的码字流和扩充后的字典，并和实验预习中的计算的结果进行比较。
2. 编码流程中需要建立初始字典，其中包含用户输入的所有单个字符。请通过更改编码流程和代码，实现无初始字典时，也能正常进行编码。

五、 实验思考题

1. 按照 LZW 编码流程，在哪些情况下需要输出 P 中的字符（或字符串）在字典中对应的编号，构成编码后的码字流？
2. 试列举一个只包含’a’，’b’，’c’，’d’四种字符的待编码字符串，要求待编码字符串占用的存储空间为 10bytes，经过 LZW 编码后得到的码字占用的存储空间为 7bytes。（假设字符串中每个单个字符分别占用 1byte 存储空间，字典中每个元素分别占用 1byte 存储空间。）