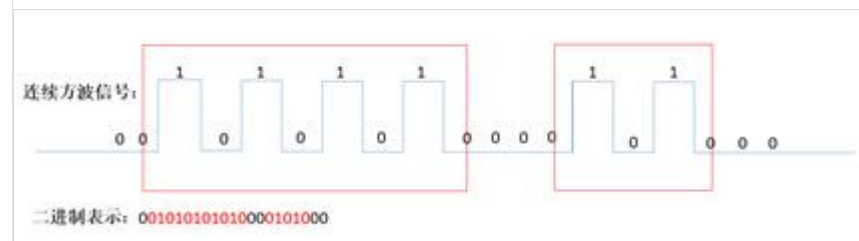


Python-题目描述:

输入一串方波信号，求取最长的完全连续交替方波信号，并将其输出，如果有相同长度的交替方波信号，输出任一即可，方波信号高位用 **1** 标识，低位用 **0** 标识，如图：



说明:

- 1) 一个完整的信号一定以 **0** 开始然后以 **0** 结尾，即 **010** 是一个完整信号，但 **101**, **1010**, **0101** 不是
- 2) 输入的一串方波信号是由一个或多个完整信号组成
- 3) 两个相邻信号之间可能有 **0** 个或多个低位，如 **0110010**, **011000010**
- 4) 同一个信号中可以有连续的高位，如 **01110101011110001010**，前 **14** 位是一个具有连续高位的信号
- 5) 完全连续交替方波是指 **10** 交替，如 **01010** 是完全连续交替方波，**0110** 不是

输入描述:

输入信号字符串（长度 ≥ 3 且 ≤ 1024 ）:

0010101010110000101000010

注：输入总是合法的，不用考虑异常情况

输出描述:

输出最长的完全连续交替方波信号串:

01010

若不存在完全连续交替方波信号串，输出 **-1**

补充说明：

输入信号串中有三个信号：**0010101010110**(第一个信号段)**0001010**(第二个信号段)**010**(第三个信号段)

第一个信号虽然有交替的方波信号段，但出现了**11**部分的连续高位，不算完全连续交替方波，在剩下的连续方波信号串中**01010**最长

示例 1

输入：

00101010101100001010010

输出：

01010

说明：

```
s = input()
last = s[0]
s1 = s[0]
MAX = ""
for i in range(1, len(s)):
    if s[i] != last:
        s1 += s[i]
        last = s[i]
    else:
        if len(s1) > 2 and s1[0] == '0' and s1[-1] == '0' and s[i] == '0':
            if len(s1) > len(MAX):
                MAX = s1
        s1 = s[i]
        last = s[i]
if len(s1) > 0 and s1[0] == '0' and s1[-1] == '0':
    if len(s1) > len(MAX):
        MAX = s1
if len(MAX) > 2:
    print(MAX)
else:
    print('-1')
```