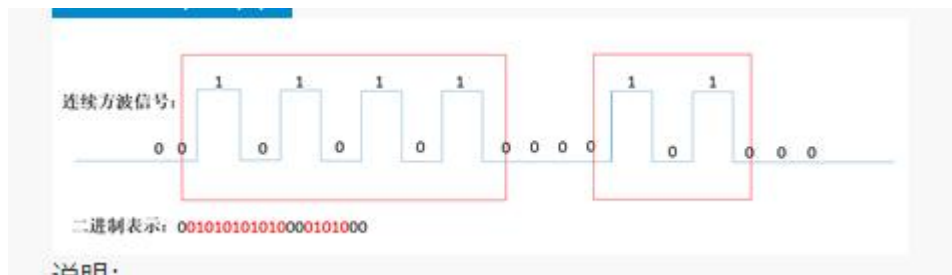


Java-滑动-输入一串方波信号

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：0 010101010110(第一个信号段) 00 01010(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:

```
import java.util.Scanner;
```

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```
public class Main {  
    public static void main(String[] args) {  
        test3();  
    }  
  
    public static void test3() {  
        Scanner in = new Scanner(System.in);  
        String signal = in.next();  
  
        int maxLength = 0;  
        int start = 0, end = 0;  
  
        int i = 0;  
        int j = 1;  
        int flag = 1;  
        char pre = signal.charAt(i);  
        while (j < signal.length()) {  
            while (!(pre == '0' && signal.charAt(j) == '0') && j < signal.length() - 1) {  
                if (pre == signal.charAt(j))  
                    flag = -1;  
                pre = signal.charAt(j);  
                j++;  
            }  
            if (maxLength < j - i + 1 && flag == 1 && j > i + 1){  
                maxLength = j - i + 1;  
                start = i;  
                end = j;  
            }  
  
            i = j;  
            pre = signal.charAt(i);  
            j++;  
            flag = 1;  
        }  
  
        if (maxLength == 0)
```

```
        System.out.println(-1);  
    else if (end == signal.length() - 1)  
        System.out.println(signal.substring(start));  
    else System.out.println(signal.substring(start,end));
```

```
    }  
}
```