# 题目

给你一个整数数据流，请你实现一个数据结构，检查数据流中最后 k 个整数是否等于给定值 value 。

请你实现 DataStream 类：

DataStream(int value, int k) 用两个整数 value 和 k 初始化一个空的整数数据流。

boolean consec(int num)将num添加到整数数据流。如果后k个整数都等于value，返回true，否则返回false。如果少于k个整数，条件不满足，所以也返回false。

示例 1：

输入：

["DataStream", "consec", "consec", "consec", "consec"]

[[4, 3], [4], [4], [4], [3]]

输出：

[null, false, false, true, false]

解释：

DataStream dataStream = new DataStream(4, 3); // value = 4, k = 3

dataStream.consec(4); // 数据流中只有 1 个整数，所以返回 False 。

dataStream.consec(4); // 数据流中只有 2 个整数

// 由于 2 小于 k ，返回 False 。

dataStream.consec(4); // 数据流最后3个整数都等于value，所以返回True。

dataStream.consec(3); // 最后 k 个整数分别是 [4,4,3] 。

// 由于 3 不等于 value ，返回 False 。

提示：

1 <= value, num <= 109

1 <= k <= 105

至多调用consec次数为105次。

# 分析

思路：

1、cnt表示末尾连续value的元素个数

2、cnt>=k, 返回true, 否则false

代码：

class DataStream {

public:

DataStream(int value, int k) {

this->\_value = value;

this->\_k = k;

}

bool consec(int num) {

if (num == \_value) {

\_cnt++;

} else {

\_cnt = 0;

}

return \_cnt >= \_k;

}

private:

int \_value, \_k;

int \_cnt = 0;

};

/\*\*

\* Your DataStream object will be instantiated and called as such:

\* DataStream\* obj = new DataStream(value, k);

\* bool param\_1 = obj->consec(num);

\*/