## 题目描述：

请实现一个函数用来找出字符流中第一个只出现一次的字符。例如，当从字符流中只读出前两个字符"go"时，第一个只出现一次的字符是"g"。当从该字符流中读出前六个字符“google"时，第一个只出现一次的字符是"l"。

如果当前字符流没有存在出现一次的字符，返回#字符。

## 解题思路

建立一个256大小的哈希表，是因为：首先，建立一个128大小的哈希表即可。然后解释：字符在计算机中以ASCII码的形式存储，当字符作为数组下标时，其表示的下标值为该字符的ASCII码的十进制值。而ASCII码由8位表示，只有后面7位起作用，2^7为128，因此ASCII码共有128个字符，十进制范围为0~127.字符串s记录了每个字符出现的顺序，hash[]数组记录每个字符出现的次数。

算法描述：

class Solution

{

public:

string s;

int hash[256] = {0};

//Insert one char from stringstream

void Insert(char ch)

{

s+=ch;

hash[int(ch)]++;

}

//return the first appearence once char in current stringstream

char FirstAppearingOnce()

{

int size = s.size();

for (int i=0; i<size;i++)

{

if (hash[s[i]] == 1)

return s[i];

}

return '#';

}

};