## 题目描述：

## 数组中有一个数字出现的次数超过数组长度的一半，请找出这个数字。例如输入一个长度为9的数组{1,2,3,2,2,2,5,4,2}。由于数字2在数组中出现了5次，超过数组长度的一半，因此输出2。如果不存在则输出0。

## 解题思路：

采用阵地攻守的思想：  
第一个数字作为第一个士兵，守阵地；count = 1；  
遇到相同元素，count++;  
遇到不相同元素，即为敌人，同归于尽,count--；当遇到count为0的情况，又以新的i值作为守阵地的士兵，继续下去，到最后还留在阵地上的士兵，有可能是主元素。  
再加一次循环，记录这个士兵的个数看是否大于数组一般即可

## 算法描述：

class Solution {

public:

int MoreThanHalfNum\_Solution(vector<int> numbers) {

int n = numbers.size();

int number = numbers[0];

int count = 1,i;

if (n==0)

return 0;

for (i=1;i<n;i++)

{

if (number == numbers[i])

count++;

else

count--;

if (0 == count)

{

number = numbers[i];

count = 1;

}

}

count = 0;

for (i=0;i<n;i++)

if (number == numbers[i])

count++;

if (count\*2 > n)

return number;

return 0;

}

};