## 题目描述：

## HZ偶尔会拿些专业问题来忽悠那些非计算机专业的同学。今天测试组开完会后,他又发话了:在古老的一维模式识别中,常常需要计算连续子向量的最大和,当向量全为正数的时候,问题很好解决。但是,如果向量中包含负数,是否应该包含某个负数,并期望旁边的正数会弥补它呢？例如:{6,-3,-2,7,-15,1,2,2},连续子向量的最大和为8(从第0个开始,到第3个为止)。给一个数组，返回它的最大连续子序列的和，你会不会被他忽悠住？(子向量的长度至少是1)解题思路

## 算法描述：

class Solution {

public:

int FindGreatestSumOfSubArray(vector<int> array) {

int currsum = array[0];

int maxsum = array[0];

int i;

for (i=1;i<array.size();i++)

{

currsum = currsum + array[i];

if (currsum < array[i])

currsum = array[i];

if (currsum > maxsum)

{

maxsum = currsum;

}

}

return maxsum;

}

};