题意：

最大子列和问题。

分析：

经典的动态规划问题。对于数列数列A[0]，A[1]，…，A[n]，我们定义函数SubSeqSum(A[i])：以A[i]结尾的最大子列和。

那么它的解可由其子问题的最优解递归定义：

SubSeqSum(A[i]) = max{0, SubSeqSum(A[i - 1])} + A[i]

换言之，在求A的最大子列和的过程中。SubSeqSum(A[i - 1])大于0则取，小于0则舍(参见L14-L17)。

本程序应注意如下细节：

1. 是否需要buffer来存储来自stdin的数据：

本身就是个O(n)的算法，且题目并未要求输出拥有最大和的子列，只需输出头尾即可，故不需额外buffer。但是需要两组数据分别记录“当前遍历”和“整个序列”的最大子列和及其头尾(sum, head和max\_sum, max\_head, max\_tail)。

1. 题目要求全负序列应输出整个序列的头尾：

将max\_head初始化全序列的第一个元素,当且仅当SubSeqSum(A[i]) >= 0时第一次修改max\_head(利用特殊值标识)，显然全负序列不满足条件，这样max\_head不会丢失。（参见L7 L19-L22）。全负时也是通过max\_sum初始的特殊值标识状态（参见 L7，L26—L29）。

3.对于拥有最大子列的子列其前部可能存在若干0，或<1,2,-3,…>的形式。故更换head的前提是sum < 0(参见L14-L17)。

4.max\_sum = INT\_MAX有两种可能性：要么全负，要么有且仅有一个正数，故需判断具体是上述那种情形(参见L27)