2006/2007 ACM International Collegiate Programming Contest   
University of Ulm Local Contest

# Problem G: Wine trading in Gergovia

Source file: gergovia.(c|cc|hs|java|pas)   
Input file: gergovia.in

As you may know from the comic "Asterix and the Chieftain's Shield", Gergovia consists of one street, and every inhabitant of the city is a wine salesman. You wonder how this economy works? Simple enough: everyone buys wine from other inhabitants of the city. Every day each inhabitant decides how much wine he wants to buy or sell. Interestingly, demand and supply is always the same, so that each inhabitant gets what he wants.

There is one problem, however: Transporting wine from one house to another results in work. Since all wines are equally good, the inhabitants of Gergovia don't care which persons they are doing trade with, they are only interested in selling or buying a specific amount of wine. They are clever enough to figure out a way of trading so that the overall amount of work needed for transports is minimized.

In this problem you are asked to reconstruct the trading during one day in Gergovia. For simplicity we will assume that the houses are built along a straight line with equal distance between adjacent houses. Transporting one bottle of wine from one house to an adjacent house results in one unit of work.

#### Input Specification

The input consists of several test cases.  
Each test case starts with the number of inhabitants *n* (2 ≤ *n* ≤ 100000). The following line contains n integers ai (-1000 ≤ ai ≤ 1000). If ai ≥ 0, it means that the inhabitant living in the ith house wants to buy ai bottles of wine, otherwise if ai < 0, he wants to sell -ai bottles of wine. You may assume that the numbers ai sum up to 0.  
The last test case is followed by a line containing 0.

#### Output Specification

For each test case print the minimum amount of work units needed so that every inhabitant has his demand fulfilled. You may assume that this number fits into a signed 64-bit integer (in C/C++ you can use the data type "long long", in JAVA the data type "long").

#### Sample Input

5

5 -4 1 -3 1

6

-1000 -1000 -1000 1000 1000 1000

0

#### Sample Output

9

9000