

Southeastern European Regional Programming Contest
Bucharest, Romania
October 18, 2014

Problem A
Banks

Input File: A.in
Output File: standard output
Program Source File: A.c, A.cpp, A.java

On Wall Street from Wonderland, we have n banks, with $10000 > n > 0$. Each bank has exactly two neighbours, the left (*L*) and right (*R*) neighbour. The first bank's left neighbour is the last bank, while the last bank's right neighbour is the first bank. Each bank $i\ (n>i\geq 0)$ has a capital k_i with $32000>k_i>-32000$. The entire capital of all banks put together is known to be positive. Whenever some capital k_i of bank i is negative, the Bank Fairy can do a magic move and turn the capital into a positive one. For instance, if $k_i=-7$, after the magic move, $k_i=7$. Unfortunately, the magic move has consequences for both neighbours of bank i. Each sees its capital reduced with the absolute value of the capital of bank i. For instance if bank *L* has capital $k_L=5$ and bank *R* has capital $k_R=11$, then after the magic move $k_L=-2$ and $k_R=4$.

Which is the minimal number of magic moves which the Bank Fairy has to do in order to make the capital of all banks greater than or equal to 0?

On the first **input** line, we have the number n of banks. On the second input line, we have the capitals $k_i\ (n>i\geq 0)$ of all banks, in the order in which they are found on Wall Street from Wonderland. Each capital is separated by a single whitespace from the next one, except for the final capital which is directly followed by the newline character.

The **output** contains a single line with the value of the minimal number of magic moves.

Sample input	Sample output
4 **1 -2 -1 3**	**9**