

Problem: The Coin Game

Input: coins.in

Program: coins.{c, cpp, java}

Alice and Ben play a simple coin game, as follows: A collection of coins of various denominations are laid out in a row. Alice plays first, removing a coin from one of the ends. Then Ben removes a coin from one end of the remaining row, and so on. Coins can be removed from either end of the row, but only from the ends. This continues until all coins have been chosen. Each player's score is the value of all the coins they collected, minus the value of the coins the other player collected. The player with the higher total wins the game. Given a row of coins, what is the best score Alice can achieve if Ben plays as well as possible?

Note that a simple greedy strategy of selecting the largest coin available will not always work. For example, if the coins' value on Alice's turn are:

5 25 10 1

In this case, Alice should select the coin of value 1. Doing so ensures that she can take the coin worth 25 on her next turn. (Ben must take the 5, or the 10; either way, Alice can take the 25).

You are given several sets of input. Each begins with an integer N ($0 < N \leq 150$) specifying how many coins are in that game. This is followed by the N values of the coins in that game. Values will not necessarily correspond to values used by U.S. (or any other) currency system, but will always be integers greater than 0 and less than 100. Input values are separated by whitespace; do not assume input is line oriented. The end of input is marked by a value of 0 for N .

You should assume that at each move of the game, each player will select the best move they can. Output for each case is a single integer: the expected amount by which the first player will win. If the first player will lose, this number will be less than 0.

Sample Input	Sample Output
4	11
5 25 10 1	9
4 3 9 12 15	0
2	-15
36 36	
3	
5 25 5	
0	