Problem 14: Optimal Coin Game

Source file: `coins.(cpp|java)`
Input file: `coins.in`
Output file: `coins.out`

Consider a single row of n coins with values $v_1 v_2 \ldots v_n$, where is n is even and $n \geq 2$. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of that coin. The object of the game is claim the largest amount of money.

The Problem

Given the values of coins that are arranged in a single row, determine the maximum possible amount of money we can ***definitely*** win assuming the opponent plays optimally to keep our score as low as possible. We have the first move.

Input File (`coins.in`)

The input file contains several test cases, one per line. Each line contains 1 or more non-negative integers, separated by a space. The first integer on a line, n (where $0 \leq n \leq 1000, n \text{ is even}$), is the number of coins in the current test case. The rest of the line will contain n integers, v_1 v_2 ... v_n (where $0 \leq v_i \leq 100$), which are the values of the n coins, listed in the same order the coins are arranged for the game.

A value of 0 for n indicates the end of the file.

Output (`coins.out`)

For each test case, output the maximum possible amount of money we can win if we play first and the opponent plays optimally, trying to make our total as low as possible.

Sample Input File

```
2 5 10
4 10 25 1 5
6 0 5 5 8 9 8
8 0 0 0 0 0 100 0 0
6 100 99 98 97 96 95
6 95 96 100 100 96 95
2 0 0
0
```

Sample Output

```
10
30
21
100
294
291
0
```