

FLATTEN

PROBLEM

A solitaire game is played given a row of N piles where each pile contains zero or more chips. See Figure 1. Piles are identified by integers 1 through N . In a *move* of this game you point to a pile, say p , and specify a number, say m . Then m chips are transferred from the pile p to each of its neighboring piles. See Figure 2 for an example.. Pile p has two neighbors, $p-1$ and $p+1$ if $1 < p < N$, and the neighbor 2 if $p=1$, and the neighbor $N-1$ if $p=N$. Note that to be able to make such a move the pile p must have at least $2m$ chips if it has two neighbors, and it must have at least m chips if it has only one neighbor.

The object of the game is to "flatten" all piles, that is make them have the same number of chips, by making as small as possible number of moves. In case there is more than one solution you are required to report only one of them.

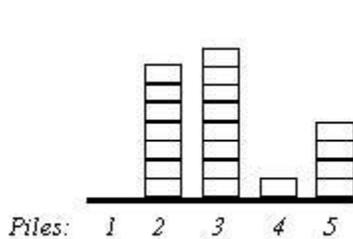


Figure 1 Five piles with 0, 7, 8, 1 and 4 chips.

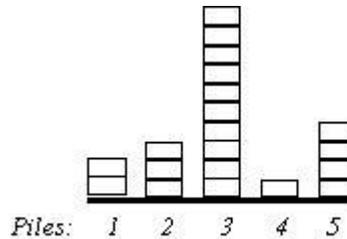


Figure 2 Configuration of same piles after a move: $p=2$, $m=2$.

ASSUMPTIONS

- It is guaranteed that it is possible to flatten the given piles in at most 110,5 000 moves.
- $2 = N = 200$
- $0 = C_i$ number of chips in a pile = 2000 where C_i is the number of chips in the pile i when the game starts ($1 = i = N$).

INPUT

The input is a text file named `flat.inp` having two lines.

- The first line: contains an integer giving the value of N .
- On the second line: there will be N integers the i th of which is the value of C_i . nonnegative integers giving the number of chips in each pile. These integers will be separated with one blank. The i th number on the second line will be the number of chips in the pile i ($1 = i = N$)

OUTPUT

The output will be the text file named `flat.out`.

- The first line: must contain the number of moves. (Call this value, call it M .)
- Each of the following M lines contains two integers representing a move: p, m . There must be M lines following the first line, each containing two integers, separated by one blank, representing a move. The first integer designates the id-number of the pile and the second integer represents the number specified in the move.

The order of moves on the output must be the same order that the moves are performed. So, your first move shall be written to the second line of the output file.

EXAMPLE

flat.inp:

```
5
0 7 8 1 4
```

flat.out

```
5
5 2
3 4
2 4
3 1
4 2
```

NOTE ON EVALUATION

Your program will be allowed to run 3 seconds.

To get full credit, A , for a test case, your number of moves, x , must be less than or equal to the number B , set by the evaluation program. Note that B is not necessarily minimal. In fact, B is chosen for a test case depending on the number of moves made by following a simple strategy with no redundant moves and the average number of chips in the piles. You can obtain partial credit for a test case. The points you get is calculated by rounding to the nearest integer the value obtained by the following formula:

$$A \quad \text{if } x < B$$

$$2 * A * (2/3 * B - x) / B \quad \text{if } B < x < 2/3 * B$$

$$0 \quad \text{if } x > 3/2 * B$$