ACM-ICPC Live Archive

4117 Postmaster’s Wish

A post master wishes to automate the sorting operation at his post office, since he finds that on an average the number of people assigned to the sorting operation are either heavily loaded or idle, ultimately resulting in loss of man hours that could have been efficiently used. He comes up with an idea of an automatic sorting machine that has a fixed capacity to hold the input letters and that sorts the letters at a fixed output rate. Depending upon the arrival of letters to the sorter, once the sorter is full of letters, the excess letters that would cause the sorting box to overflow have to be dropped into a bin to be manually sorted afterwards.

You are working for a company that has been called by the postmaster for a meeting to demonstrate a software simulation of the sorter. The following scenarios need to be shown in the simulation to him:

- a) Sorter box is completely filled.
- b) Sorter box is drained out.
- c) Sorter box overflows.

At the beginning of every second:

- A certain number of messages arrive, and they are added to the sorter bin of a fixed capacity C. Excess messages cause overflow into the manual sorter bin.
- From the letters that are pending to be sorted, the sorter acts and sorts R letters.

Input

Input consists of multiple testcases.

The first line of each testcase contains two integers: C, R, M ($1 \leq M \leq 100$; $1 \leq R \leq C \leq 20,000$) representing the capacity of the sorter, the fixed-rate of sorting of the sorter and the number of messages.

The following M lines contain one integer each, i-th line representing the number of letters that arrived, at time i. Each of these integers will be in the inclusive range $[0, 1000]$.

Input ends with a line containing three zeros.

Output

Output to each test case must begin with a line containing ‘`Case #`$< case - number >$`:`’

Each test-case must contain M lines, with exactly 5 integers each, which the state of the sorter after every unit of time. The five integers represent the following:

- Current time (The i-th line will be just i).
- Letters received (Same as the input for the i-th line).
- Letters sorted (Number of letters the sorter sorted at this unit of time; must be less than or equal to R).
- Sorter balance (Number of pending letters in the sorter; must be less than or equal to C).
- Number of letters dumped into the manual sorter bin.

There should be a blank line after each test-case.

Sample Input

```
10 7 3
12
3
2
876 321 7
500
200
900
300
0
1
1
0 0 0
```

Sample Output

```
Case #1:
1 12 7 3 2
2 3 6 0 0
3 2 2 0 0

Case #2:
1 500 321 179 0
2 200 321 58 0
3 900 321 555 82
4 300 321 534 0
5 0 321 213 0
6 1 214 0 0
7 1 1 0 0
```