Enrich my purse

Jack plays this ball game for the first time in his club. Jack has a ball, which bounces with a width of **W**. Coins are arranged on a straight line at regular intervals. If the ball strikes the i-th coin, Jack gains **money**[i] (which could possibly be negative). Jack can take at most **B** turns, to throw the ball. At each turn, jack can either throw the ball from left to right, or right to left, and choose which ball to start the knock out. If he chooses to knock out from ball i to the right, he will knock out i, i+W, i+2W ... Similarly if he chooses to knock out from right to left, starting from ball i he will knock out, i, i-W, i-2W ... Please note that once a ball is knocked out, it is removed and it's place contains a void. i.e., you cant gain **money[i]** for the same **i** twice.
Jack wants to maximise his money gained, by carefully choosing his turns. If there is more than one way to gain the same money, jack wishes to minimise the number of times he throws.

Input Format

The input file consists of multiple testcases.
The first line of each testcase contains three integers, **W B N** (1 <= N <= 100; W,B > 0)
The second line of each testcase contains N integers, denoting **money[i]**. (| money[i] | <= 10^6)
Input terminates with a line containing three zeros which must not be processed.

Output Format

For each testcase print one line denoting the maximal money gained and the number of turns taken. Please see the sample output and stick to the output format.
"Case#id: Jack wins $X out of Y throws."
NOTE: You must spell the same way the sample output says. Extra spaces and case insensitivity can cause wrong answer responses.

Test data:
100 test cases, Time limit: 10s

Sample Input:

```
2 3 10
-1 3 2 5 1 -2 0 5 1 -3
2 3 14
-1 3 2 5 -5 -5 1 -2 0 5 -5 -5 1 -3
3 3 5
-1 -2 -3 -4 -5
1 2 6
-1 -1 10 10 -1 -1
0 0 0
```

Sample Output:

```
Case#1: Jack wins $15 out of 2 throws.
Case#2: Jack wins $10 out of 3 throws.
Case#3: Jack wins $0 out of 0 throws.
Case#4: Jack wins $18 out of 1 throws.
```

Output Explanation:

We present one of the optimal solutions. We number balls from 1 to N.

TestCase#1: [Jack takes only two throws, though he can take three]

Throw#1: From ball#3 towards right, 2 + 1 + 0 + 1 = 4

Throw#2: From ball#8 towards left, 5 + -2 + 5 + 3 = 11

TestCase#2:

Throw #1: From ball#3 towards left, 2 + -1 = 1

Throw #2: From ball#4 towards left, 5 + 3 = 8

Throw #3: From ball#13 towards right, 1 = 1

TestCase#3:

All numbers are negative. Jack takes no throws.