R-P-S Tournament (`rock-paper-scissors-tournament`)

There are M tram stations arranged in a circle, numbered from 1 to M. There are N players, numbered from 0 to $N-1$, who wish to participate in a round-robin rock-paper-scissors tournament. Each player aims to play against every other player exactly once. However, two players can only play if they are at the same station.

Figure 1: One of the stations.

At the start, each player selects a station and purchases a ticket that permits them to move either clockwise or counterclockwise. The ticket remains valid throughout the tournament.

Each minute, all players simultaneously move to the next station in the direction specified by their ticket. This movement happens instantaneously around the circle. Following a movement (or initially), players who meet at the same station engage in a match. If two players meet again later, they do not play another match.

You are required to answer two questions:

A) How many matches will be played in total?

B) At what time does the last possible match occur?

> ☞ Among the attachments of this task you may find a template file
> `rock-paper-scissors-tournament.*` with a sample incomplete implementation.

Input

The input consists of:

- A line containing integers N, M; representing the number of players and the number of stations.

- A second line containing N integers $S_0, \ldots, S_{N-1}$, where $|S_i|$ denotes the starting station of player i. If S_i is positive, the player travels clockwise, and if S_i is negative, the player travels counterclockwise.

Output

The output should contain a single line with two integers:

- G: the total number of matches played.

- T: the number of minutes until the last possible match occurs.

Here, minute i corresponds to the i-th movement of the players to the next station. Matches at the initial stations happen in minute 0, and the first movement happens in minute 1.

If no matches are played, output `"0 -1"` (without quotes).

Constraints

- $1 \le N \le 100\,000$.

- $3 \le M \le 10^9$.

- $1 \le |S_i| \le M$ for each $i = 0 \ldots N - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (18 points) $N \le 100$, $M \le 100$.

– **Subtask 3** (8 points) $N \le 1000$, $M \le 1000$.

– **Subtask 4** (14 points) $N \le 10\,000$, $M \le 10\,000$ and M is even.

– **Subtask 5** (16 points) $N \le 10\,000$, $M \le 10\,000$ and M is odd.

– **Subtask 6** (10 points) $N \le 100\,000$, $M \le 100\,000$.

– **Subtask 7** (34 points) No additional limitations.

In each subtask, you can earn a **partial score**. You will receive 50% of the points for a subtask if you correctly determine the total number of matches played in every test case, but make an error in determining the number of minutes in at least one case.

To earn the partial score, ensure that your program's output conforms to the output specification. Failure to do so may result in the grader rejecting your solution due to formatting errors. Specifically, your program must output two integers, where the second integer can be any value if the first integer (the number of matches) is correct.

Examples

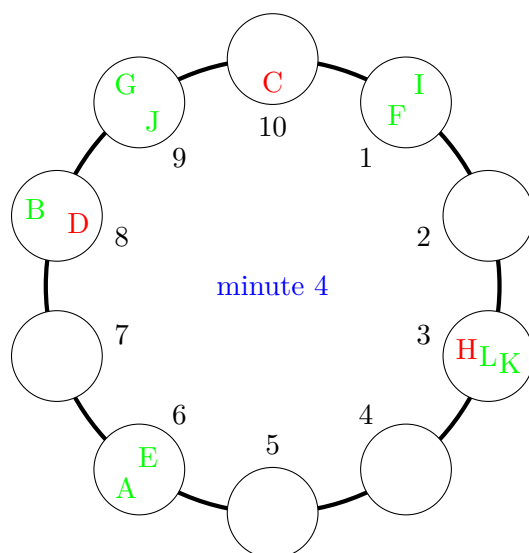| input	output
12 10 2 4 -4 -2 2 7 5 -7 7 5 9 9	16 4

---

input	output
4 5 2 2 2 -2	6 0
2 5 2 3	0 -1

Explanation

In the **first sample case** denote the the players by A, B, C, D, E, F, G, H, I, J, K, L. The figure below (red players go counterclockwise, green players go clockwise) shows that:

- In minute 0 we have 9 games.

- In minute 1 we have 4 games.

- In minute 2 we have 0 games.

- In minute 3 we have 0 games.

- In minute 4 we have 3 games.
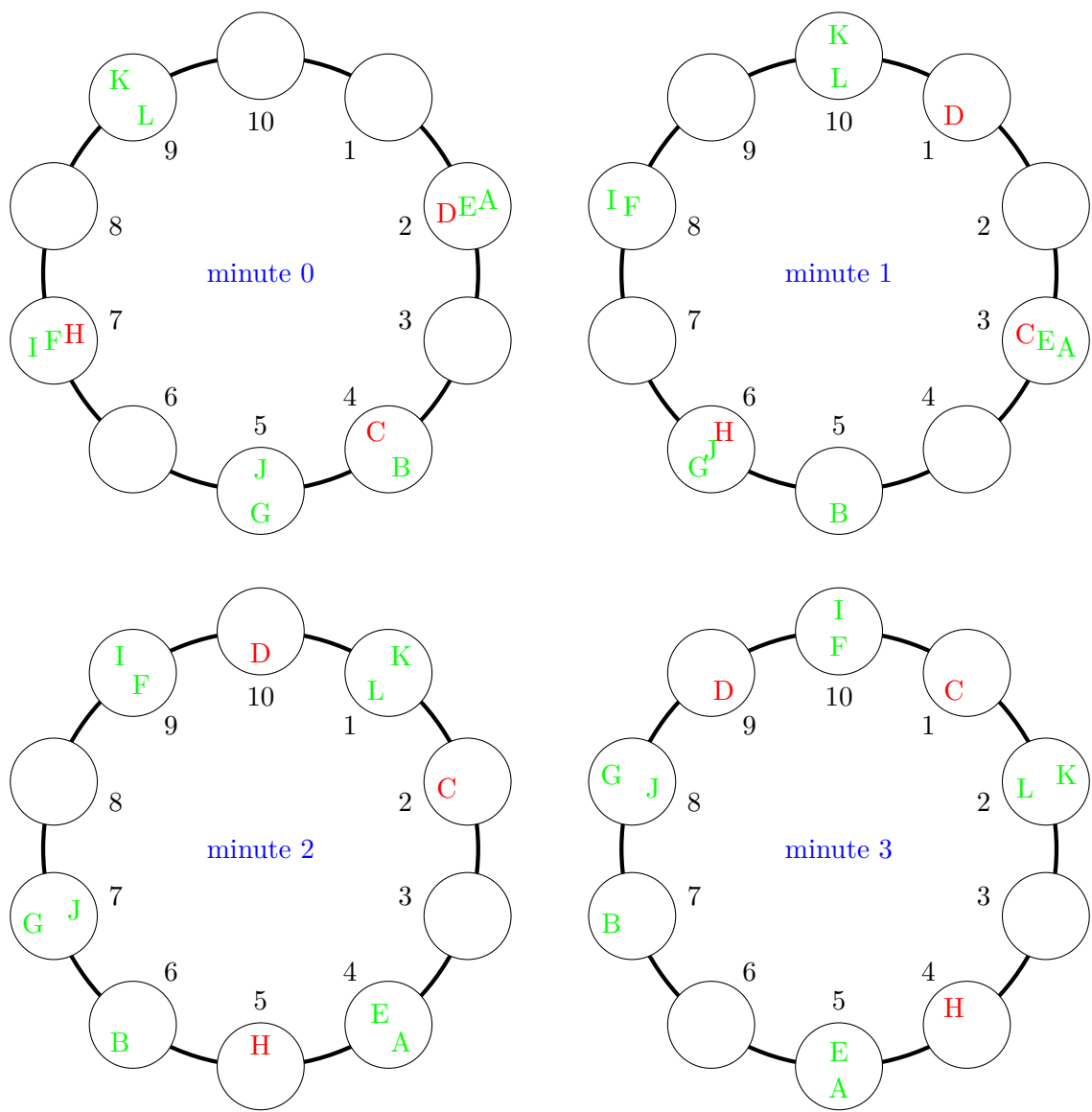
minute 4

It can be proven that no more games occur after minute 4.

In the **second sample case** everyone starts at the same station, so all games are played in minute 0.