ChannelTalk

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

ChannelTalk is an all-in-one AI messenger developed by Channel Corp, helping companies achieve sustainable development through real-time communication with their customers. By utilizing CRM data and AI, it enhances efficiency and improves customer experience, delivering service that stays true to the philosophy of 'the customer holds the answer.' ChannelTalk has secured a market-leading share in Japan and is rapidly increasing its revenue, planning to expand beyond Asia into the US market. The key to success lies in the 'product'—over half of ChannelTalk's employees are developers, all focused as a team on creating one outstanding product.

To gather customer opinion, Channel Corp. has decided to pilot several **discussion channels** on ChannelTalk, its communication platform.

There are N channels, numbered from 1 to N. All channels discuss opposing opinions on the same topic. In channel i, there are A_i people who agree with the opinion and B_i people who disagree. Additionally, the maximum capacity of channel i is C_i. C_i is an even number. (As an exception, channel N has no limit on the number of participants.)

Sometimes, a new participant enters a channel. This participant also either agrees or disagrees with the opinion. Each time a new participant enters, the following procedure occurs based on ChannelTalk's policy.

- When a new participant enters a channel, if the number of participants does not exceed the capacity of the channel, they stay in the channel.
- If the number of participants has exceeded the channel's capacity, the total number of participants becomes an odd number. In this case, one participant from the side with more people, either those who agree or disagree, is automatically moved to the channel with the next number. (The opinion of the participant who was moved remains unchanged.)
- If the number of participants in the next channel exceeds the channel's capacity as a result of the above step, the same process is repeated. This process is repeated until the number of participants in each channel is within its capacity. Note that there is no capacity limit on channel N, so this process eventually terminates.

To assist in piloting this feature, you need to write a program that manages the number of participants in each channel as new participants enter. Specifically, your program needs to perform the following queries quickly.

- `1 x v`: v new participants who agree with the opinion enter channel x.
- `2 x v`: v new participants who disagree with the opinion enter channel x.
- `3 x`: Print the number of participants who agree and the number of participants who disagree in channel x.

Note that in queries of type 1 and 2, the v new participants enter the channel one by one, and the next participant enters once all channel reallocation processes due to the previous participant are completed.

Input

The first line of input contains two space-separated integers — N and Q. ($2 \leq N \leq 200\,000$; $1 \leq Q \leq 200\,000$)

The i-th of the following $N-1$ lines contains three space-separated integers — A_i, B_i, and C_i. ($0 \le A_i, B_i \le 10^9$; $2 \le C_i \le 10^9$; $A_i + B_i \le C_i$; C_i is even)

The $N+1$-th line of the input contains two space-separated integers — A_N and B_N. ($0 \le A_N, B_N \le 10^9$)

The i-th of the following Q lines describes the i-th query in the same format as given in the problem statement. For all queries, the following conditions are satisfied: $1 \le x \le N$, $1 \le v \le 10^9$.

It is guaranteed there is at least one query of type 3.

Output

For all queries of type 3, print the number of participants who agree and the number of participants who disagree in the corresponding channel.

Examples

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