Locomotive Lane Logistics

Dooby is an energetic and meticulous train enthusiast, recently hired as a train operator. However, her first assignment is far from simple.

Dooby is responsible for managing a new train track connecting *Primstation* and *Krustown*. The train track is a straight line, with a total length of s meters. Each day, Dooby must create a schedule for the trains running on this track, and the list of trains varies day by day.

Initially, there are no trains. Dooby's job spans q days, and on the i-th day, one of the following events occurs:

- "$+\ w_i\ v_i$" – a train with a length of w_i meters and a constant speed of v_i meters per hour is **added** to Dooby's train list.
- "$-\ w_i\ v_i$" – a train with a length of w_i meters and a constant speed of v_i meters per hour is **removed** from Dooby's train list. It is guaranteed that at least one such train exists in the current list.

Each day, all trains in Dooby's list start at *Primstation* and must travel to *Krustown*. Since there is only one track, the trains must be arranged in a specific order and start at specific times to **avoid collisions**. Once a train reaches *Krustown*, it is **immediately** moved to a shed, allowing the next train to proceed.

Before departure, each train's **front** end is aligned with the start of the track. A train is considered to have reached *Krustown* when its **rear** end crosses the far end of the track.

To demonstrate her efficiency as a train operator, Dooby must find the starting times for all trains each day so that:

- there must be no collision between two trains, and
- the time t_i required for all trains to reach *Krustown* is **minimized**.

Your task is to assist Dooby in her scheduling mission. For each of the q days, please determine the **minimum** possible time t_i required for all trains currently in the list to safely reach *Krustown*. Output t_i modulo $998\,244\,353$.

Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Collision condition

For the i-th train, let $\text{Rear}_i(t)$ and $\text{Front}_i(t)$ represent the positions of the **rear** end and the **front** end of the i-th train at time t, respectively.

By definition, we have $\text{Front}_i(t) = \text{Rear}_i(t) + w_i$.

Two trains i and j are considered to have **collided** if there exists a moment t such that:

- both the i-th and j-th trains have left *Primstation*,
- neither the i-th nor the j-th train has reached *Krustown*,
- $\min\{\text{Front}_i(t), \text{Front}_j(t)\} > \max\{\text{Rear}_i(t), \text{Rear}_j(t)\}$

Input

The first line contains two integers s and q ($1 \leq s \leq 10^9$, $1 \leq q \leq 200\,000$) – the length of the track in meters and the number of days Dooby's job lasts.

Each of the following q lines describes an event on the i-th day. It is one of the following types:

- "$+\ \ w_i\ v_i$" ($1 \leq w_i, v_i \leq 10^6$) – a train with a length of w_i meters and a constant speed of v_i meters per hour is **added** to Dooby's train list.
- "$-\ \ w_i\ v_i$" ($1 \leq w_i, v_i \leq 10^6$) – a train with a length of w_i meters and a constant speed of v_i meters per hour is **removed** from Dooby's train list. It is guaranteed that at least one such train exists in the current list.

Output

Print q lines. The i-th line should contain the **minimum** possible time required for all trains in the list on the i-th day to safely reach *Krustown*, modulo $998\,244\,353$. If on the i-th day, there is no train in Dooby's list, output `0`.

Sample Input 1	Sample Output 1
20 4 + 2 2 + 3 3 + 5 2 - 2 2	11 12 499122191 499122190

Sample Explanation

Let (w, v) denote a train with length of w meters and speed of v meters per hour.

In the sample test, we have a train track of length 20 meters. Even though this is a small track, Dooby's efficient scheduling ensures every train reaches safely and quickly!

1. On the first day, there is only one train $(2, 2)$. The train takes $20/2 = 10$ hours for the **front** to reach *Krustown*, and $2/2 = 1$ hours for the rear to clear the track.
2. On the second day, a new train $(3, 3)$ was added.

   This train can start at $4\frac{1}{3}$ hours. It will reach *Krustown* at exactly 12 hours.

   Note that if the train $(3, 3)$ starts sooner, it can collide with the train $(2, 2)$. For example, suppose that the train $(3, 3)$ starts at 3 hours, the collision will happen at 7 hours.
3. On the third day, another train $(5, 2)$ was added. Here is one schedule that ensure the minimal time.
   - Train $(2, 2)$ starts at 0 hours;
   - Train $(5, 2)$ starts at 1 hours;
   - Train $(3, 3)$ starts at $6\frac{5}{6}$ hours.

   The time for the train $(3, 3)$ to reach *Krustown* is $\frac{29}{2}$ hours. The output is 499122191 because $499122191 \cdot 2 \equiv 29 \pmod{998\,244\,353}$.
4. On the forth day, the train $(2, 2)$ is removed. One optimal schedule is as follows:
   - Train $(3, 3)$ starts at 0 hours;
   - Train $(5, 2)$ starts at 1 hours.

   The time for the train $(5, 2)$ to reach *Krustown* is $\frac{27}{2}$ hours.

This page is intentionally left blank.