Reazione a candela (candele)

Monica owns an elegant collection of N candles (the tall-and-narrow kind). The candles are indexed from 1 to N and, some of them being partially consumed, they have, generally speaking, different lengths. Mojito – Monica's dog – has been chasing a tennis ball and, in the process, made all the candles fall onto the floor! Due to pure luck, the candles fell in such a way that they are now aligned on the same line (possibly overlapping). The i-th candle's fuse is located at position M_i, whereas its base is located at position B_i. Here M_i and B_i are assumed to be non-negative integers.

Initially, all candles are unlit. When lit, a candle burns at the constant rate of one unit per second. Thus, when t seconds have passed from the instant a certain candle gets lit, its fuse has moved by exactly t units (either to the left or to the right, according to the candle's orientation). As soon as the fuse of a lit candle shares the same position of the fuse of an unlit candle, the latter gets lit as well, in a cascade-like effect.

Monica decides to lit candle 0 at the time $t_0 = 0$. Now, for each candle, she wonders how many seconds after t_0 she'd need to wait for that candle to get lit, or if it will remain unlit forever. Help her out!

Implementation

You should submit a single file, with either a `.c` or `.cpp` extension.

> ☞ Among the attachments in this task you will find a template `candele.c` or `candele.cpp` with a sample implementation.

You will have to implement the following function:

C	`void brucia(int N, int *M, int *B, long long *T);`
C++	`void brucia(int N, vector<int> &M, vector<int> &B, vector<long long> &T);`

- The integer N is the number of candles.
- Array M, indexed from 0 to $N-1$, contains the positions of the candles' fuses. Specifically, $M[i] = M_i$ for all $i = 0, \ldots, N-1$.
- Array B, indexed from 0 to $N-1$, contains the positions of the candles' bases. Specifically, $B[i] = B_i$ for all $i = 0, \ldots, N-1$.

The function `brucia` should fill array T, indexed from 0 to $N-1$, with the times at which the candles get lit. If a candle remains unlit forever, the value in the array corresponding to its index should be -1. **The time at which candle 0 gets lit has to be specified as well.**

Sample grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file is made up of $N + 1$ lines, containing:

- Line 1: the integer N.
- Line $2 + i$: two space-separated integers, M_i and B_i.

The output file is made up of a single line, contaning N space separated integers: $T[0], T[1], \ldots, T[N-1]$.

Constraints

- $1 \leq N \leq 500\,000$.
- $0 \leq M_i,\ B_i \leq 10^9$ for all $i = 0,\ \ldots,\ N - 1$.
- $M_i \neq B_i$ for all $i = 0,\ \ldots,\ N - 1$.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all the testcases it contains.

In the following, the maximum of the $2N$ numbers $M_0,\ \ldots,\ M_{N-1},\ B_0,\ \ldots,\ B_{N-1}$ is denoted by L.

- **Subtask 1 [0 points]**: Sample cases.
- **Subtask 2 [6 points]**: Each candle burns towards the right (i.e. $M_i < B_i$ for all i's), and it is guaranteed that all the candles get lit in a finite amount of time (i.e. $T[i] \neq -1$ for all i's).
- **Subtask 3 [10 points]**: Each candle burns towards the right (i.e. $M_i < B_i$ for all i's).
- **Subtask 4 [9 points]**: $N,\ L \leq 50$.
- **Subtask 5 [15 points]**: $N \leq 50\,000,\ L \leq 200$.
- **Subtask 6 [17 points]**: $N \leq 5\,000$.
- **Subtask 7 [14 points]**: Candles are at most 10 units long (i.e. $|M_i - B_i| \leq 10$ for all i's).
- **Subtask 8 [29 points]**: No additional constraints.

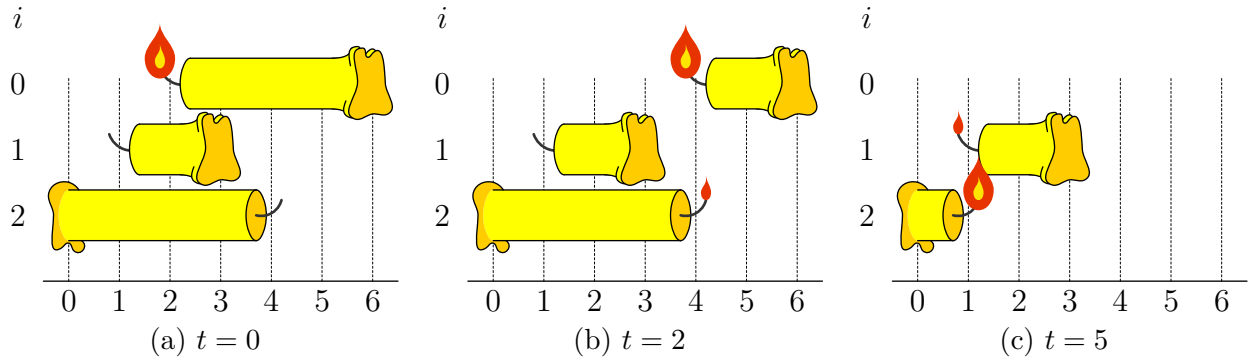
Examples

stdin	stdout
3 2 6 1 3 4 0	0 5 2
4 2 5 2 4 2 0 7 2	0 0 0 -1

stdin	stdout
8 18 14 5 12 13 22 21 10 8 9 19 12 21 19 15 20	0 -1 13 21 -1 7 21 3
4 1 999999998 999999999 1000000000 999999998 0 0 999999999	0 2999999994 999999997 1999999995

Explanation

In the **first sample case**, candle 0 starts burning towards the left at time $t = 0$. At time $t = 2$, its fuse is at position $2 + 2 = 4$, which matches the position of candle 2's fuse. Therefore, candle 2 gets lit. When another second has passed, candle 0 is extinguished, and after 2 more seconds (that is, at time $t = 5$) the fuse of candle 2 shares the same position of the fuse of candle 1, which is indeed the last to get lit.



(a) $t = 0$ (b) $t = 2$ (c) $t = 5$

In the **second sample case**, candle 0's fuse already overlaps with those of candles 1 and 2, which are immediately lit (at time $t = 0$). On the other hand, candle 3 will never be reached by the fuse of another candle, and thus it will remain unlit.