B. Processing Queries

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem you have to simulate the workflow of one-thread server. There are n queries to process, the i-th will be received at moment t_i and needs to be processed for d_i units of time. All t_i are guaranteed to be distinct.

When a query appears server may react in three possible ways:

1. If server is free and query queue is empty, then server immediately starts to process this query.
2. If server is busy and there are less than b queries in the queue, then new query is added to the end of the queue.
3. If server is busy and there are already b queries pending in the queue, then new query is just rejected and will never be processed.

As soon as server finished to process some query, it picks new one from the queue (if it's not empty, of course). If a new query comes at some moment x, and the server finishes to process another query at exactly the same moment, we consider that first query is picked from the queue and only then new query appears.

For each query find the moment when the server will finish to process it or print -1 if this query will be rejected.

Input

The first line of the input contains two integers n and b $(1 \le n, b \le 200\,000)$ — the number of queries and the maximum possible size of the query queue.

Then follow n lines with queries descriptions (in chronological order). Each description consists of two integers t_i and d_i $(1 \le t_i, d_i \le 10^9)$, where t_i is the moment of time when the i-th query appears and d_i is the time server needs to process it. It is guaranteed that $t_{i-1} < t_i$ for all $i > 1$.

Output

Print the sequence of n integers $e_1, e_2, \ldots, e_n$, where e_i is the moment the server will finish to process the i-th query (queries are numbered in the order they appear in the input) or -1 if the corresponding query will be rejected.

Examples

input
5 1 2 9 4 8 10 9 15 2 19 1
output
11 19 -1 21 22

input
4 1 2 8 4 8 10 9 15 2
output
10 18 27 -1

Note

Consider the first sample.

1. The server will start to process first query at the moment 2 and will finish to process it at the moment 11.
2. At the moment 4 second query appears and proceeds to the queue.
3. At the moment 10 third query appears. However, the server is still busy with query 1, $b = 1$ and there is already query 2 pending in the queue, so third query is just rejected.
4. At the moment 11 server will finish to process first query and will take the second query from the queue.
5. At the moment 15 fourth query appears. As the server is currently busy it proceeds to the queue.
6. At the moment 19 two events occur simultaneously: server finishes to proceed the second query and the fifth query appears. As was said in the statement above, first server will finish to process the second query, then it will pick the fourth query from the queue and only then will the fifth query appear. As the queue is empty fifth query is proceed there.
7. Server finishes to process query number 4 at the moment 21. Query number 5 is picked from the queue.
8. Server finishes to process query number 5 at the moment 22.