Task 2: Fuel Station (`fuelstation`)

As oil prices plummet, Pengu the Penguin has decided to visit Squeaky the Mouse who lives D kilometres away.

Pengu's *spheniscidae-mobile* starts its journey with F litres of fuel, consumes 1 litre of fuel per kilometre, and is able to hold any amount of fuel at any point in time.

Furthermore, there are N fuel stations between Pengu and his destination, with the i^{th} fuel station being X_i kilometres away from Pengu's house. At each fuel station, Pengu is only able to top up A_i litres of fuel (a limit imposed to prevent drivers from hoarding cheap fuel), and only if $F \leq B_i$ (to ensure that the fuel goes to drivers who most need it), Here, F refers to the amount of fuel (in litres) that Pengu **started** with.

Being an efficient penguin, Pengu would like to minimise the value of F while still being able to reach his destination.

Input

Your program must read from standard input.

The first line contains two integers N and D.

N lines will follow. The ith line contains three integers X_i, A_i and B_i, which represent the i^{th} fuel station.

Output

Your program must print to standard output.

The output should contain a single integer on a single line, the minimum value of F needed to reach the destination.

Implementation Note

As the input lengths for subtasks 2, 4, and 7 may be very large, you are recommended to use C++ with fast input routines to solve this problem. The scientific committee does not have a solution written in Python that can fully solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `FuelStation.java` and place your main function inside `class FuelStation`.

Subtasks

The maximum execution time on each instance is 3.0s, and the maximum memory usage on each instance is 1GiB. For all testcases, the input will satisfy the following bounds:

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq A_i, B_i, D \leq 10^9$
- $0 < X_i < D$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	7	$N = 1$
2	13	$B_i = 10^9$
3	17	$1 \leq D \leq 10^4, 1 \leq N \leq 10^4$
4	12	$1 \leq D \leq 10^4$
5	19	$1 \leq N \leq 16$
6	11	$1 \leq N \leq 10^4$
7	21	-

Sample Testcase 1

This testcase is valid for subtasks 1, 3, 4, 5, 6 and 7 only.

Input	Output
`1 10` `4 8 6`	`4`

Sample Testcase 1 Explanation

We start with $F = 4$ litres of fuel and

1. Reach the only fuel station ($X_1 = 4$) with $4 - 4 = 0$ litres of fuel left.
2. Top up $A_1 = 8$ litres of fuel since $F \leq B_1 = 6$ to obtain $0 + 8 = 8$ litres of fuel left.
3. Reach our destination at $X = 10$ with $8 - (10 - 4) = 2$ litres of fuel left.

This is the minimum F possible.

Sample Testcase 2

This testcase is valid for subtasks 3, 4, 5, 6 and 7 only.

Input	Output
5 100 50 30 25 50 40 25 25 25 25 75 20 25 5 5 25	20

Sample Testcase 2 Explanation

We start with $F = 20$ litres of fuel and

1. Reach the 5^{th} fuel station ($X_5 = 5$) with $20 - 5 = 15$ litres of fuel left.
2. Top up $A_5 = 5$ litres of fuel since $F \leq B_5 = 25$ to obtain $15 + 5 = 20$ litres of fuel left.
3. Reach the 3^{rd} fuel station ($X_3 = 25$) with $20 - (25 - 5) = 0$ litres of fuel left.
4. Top up $A_3 = 25$ litres of fuel since $F \leq B_3 = 25$ to obtain $0 + 25 = 25$ litres of fuel left.
5. Reach the 1^{st} and 2^{nd} fuel stations ($X_1 = X_2 = 50$) with $25 - (50 - 25) = 0$ litres of fuel left.
6. Top up $A_1 + A_2 = 70$ litres of fuel since $F \leq B_1 = B_2 = 25$ to obtain $0 + 70 = 70$ litres of fuel left.
7. Reach our destination at $X = 100$ with $70 - (100 - 50) = 20$ litres of fuel left.

This is the minimum F possible.

Note that we could have also used the 4^{th} fuel station ($X_4 = 75$) if we wanted to. Although we reach it with $70 - (75 - 50) = 45 \not\leq B_4 = 25$ litres of fuel left, we can still use it since $F = 20 \leq B_4 = 25$. However, we are still able to reach our destination even if we do not use the fuel station.