

Pizza

Hiện tại, bài tập này đã có trên online judge chính thức của VNOI, bạn có thể truy cập ở đây: <https://oj.vnoi.info/problem/vmpizza>

Linh đang sống trên tầng 5 ở East Campus, một trong những kí túc lâu đời nhất của MIT. Một điểm bất tiện của kí túc này là không có thang máy, dẫn đến sự đi lại khó khăn.

Sau một buổi làm việc căng thẳng, Linh quyết định đặt pizza thay cho bữa tối. Để thay đổi khẩu vị, cậu đặt N pizza tại N cửa hàng khác nhau. Pizza thứ i được chở đến tại thời điểm t_i , cung cấp a_i năng lượng nếu ăn ngay. Tuy nhiên, Pizza sẽ mất dần độ nóng hồi cũng như năng lượng theo thời gian: qua mỗi thời điểm, pizza thứ i mất b_i năng lượng.

Linh biết trước thời điểm và giá trị của N loại pizza, và cậu muốn đạt được càng nhiều năng lượng càng tốt. Mỗi lần, Linh có thể chạy xuống nhận pizza, mang về phòng và ăn hết số pizza trong thời gian không đáng kể. Tuy nhiên, nên nhớ rằng Linh sống ở tầng 5, mỗi chuyến đi sẽ tiêu tốn của cậu B năng lượng. Linh không thích phí phạm đồ ăn (mặc dù đã sống gần 2 năm ở Mỹ), cậu sẽ luôn ăn hết pizza, kể cả khi năng lượng của nó xuống dưới 0 (khi ăn thì Linh sẽ bị cộng thêm một lượng âm năng lượng, hay nói cách khác là bị giảm năng lượng).

Xác định lượng năng lượng tối đa Linh sẽ có với chiến thuật chạy tối ưu.

Input

Dòng 1: 2 số nguyên N và B .

Dòng 2.. $N + 1$: mỗi dòng chứa 3 số nguyên t_i, a_i, b_i ($1 \leq t_i, a_i, b_i \leq 10^5$)

Output

Một dòng duy nhất: năng lượng tối đa với chiến thuật chạy tối ưu. Kết quả nằm trong phạm vi số nguyên 64-bit (*int64* trên Pascal hoặc *long long* trên C++).

Giới hạn

- $1 \leq N \leq 10^5$.
- $1 \leq B \leq 10^5$.
- Trong ít nhất 30% số test, tương ứng với 30% số điểm, $N \leq 1000$
- Trong quá trình thi, bài của bạn chỉ được chấm với 2 test ví dụ. Kết quả sẽ hiện 100 nếu bài của bạn chạy đúng cả 2 test.

Ví dụ

Input 1:

1 4 1
2 6 1

Output 1:

4

Input 2:

2 3
1 1 100
2 10 1

Output 2:

5

Giải thích

- Trong test 1, phương án tối ưu là chạy xuống 1 lần tại thời điểm 2, lấy cả 2 pizza lên và ăn ngay. Kết quả = $(4 - 1) + (6 - 0) - 5 = 4$.
- Trong test 2, phương án tối ưu là chạy xuống 2 lần tại thời điểm 1 và 2. Kết quả = $(1 - 0) - 3 + (10 - 0) - 3 = 5$.