|  |
| --- |
| BGMINE - Xây đập giữ vàng |

Trên một con đường biểu diễn như trục số thực có n mỏ vàng đánh số từ 1 tới n. Mỏ thứ  i nằm ở tọa độ xi, có tổng trữ lượng vàng là gi, trong mỏ còn có lượng đá đủ để xây dựng đoạn kè có độ dài ri. Trong mùa mưa lũ, việc phòng chống ngập cho các mỏ trở nên cấp thiết và rất khó khăn trong việc vận chuyển vật liệu xây kè. Vì vậy, Chính phủ muốn  dùng đá ở một dãy mỏ liên tiếp để xây dựng một đoạn kè liên tục bảo vệ tất cả các mỏ đó. Ta có thể coi cửa các mỏ vàng rất nhỏ, nên dù chỉ nằm ở đầu đoạn kè thì mỏ vẫn được an toàn.

Yêu cầu: Hãy giúp chính phủ xác định đoạn kè có thể xây dựng với tổng trữ lượng vàng trong các mỏ được bảo vệ là lớn nhất.

### Input

* Dòng đầu chứa số nguyên dương  n ≤ 105  là số lượng mỏ vàng.
* n dòng tiếp theo, dòng thứ i chứa ba số nguyên xi, gi, ri cách nhau bởi dấu cách (−109  ≤ x1  < x2  < ⋯ < xn  ≤ 109; 0 ≤ gi, ri   ≤ 109)

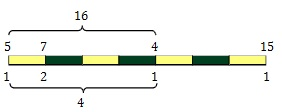
### Output

* Một số nguyên duy nhất là tổng trữ lượng vàng lớn nhất trong các mỏ được bảo vệ theo phương án tìm được.

Example

Input:

|  |
| --- |
| 4 |
| 0 5 1 |
| 1 7 2 |
| 4 4 1 |
| 7 15 1 |



Output:

16

Chú ý:

* 50% điểm ứng với các tests có n ≤ 5000
* 50% số điểm ứng với các tests có 10000 ≤ n ≤ 105

Theo yêu cầu đề bài thì mình cần tìm một số hang liên tiếp mà có tổng số đá lớn hơn hoặc bằng độ dài các hang và có tổng số vàng lớn nhất đó: Sumr[j] – Sumr[i-1] >= x[j] – x[i] với (j >= i).

Ta biến đổi điều kiện trên thành: Sumr[j] – x[j] >= Sumr[i-1] – x[i] với (j >= i).

Ta sử dụng ctdl BIT để tính nhanh hơn. Ta cần rời rạc hóa giá trị Sumr[i] – x[i] và Sumr[i-1] – x[i] để lưa được trên cây BIT.

ĐPT: O(nlogn)

Given two strings, a and b, find and print the total number of ways to insert a character at any position a in string such that the length of the Longest Common Subsequence of characters in the two strings increases by one.

Input Format

The first line contains a single string denoting a.   
The second line contains a single string denoting b .

Constraints

Scoring

1<=lengh(a),length(b)<=5000

* Strings a and b are alphanumeric (i.e., consisting of arabic digits and/or upper and lower case English letters).
* The new character being inserted must also be alphanumeric (i.e., a digit or upper/lower case English letter).

Output Format

Print a single integer denoting the total number of ways to insert a character into string in such a way that the length of the longest common subsequence of and increases by one.

Sample Input

aa

baaa

Sample Output

4

Explanation

The longest common subsequence shared by a and b is aa, which has a length of 2 . There are two ways that the length of the longest common subsequence can be increased to by adding a single character to :

1. There are 3 different positions in string where we could insert an additional a to create longest common subsequence aaa (i.e., at the beginning, middle, and end of the string).
2. We can insert a b at the beginning of the string for a new longest common subsequence of baa.

As we have 3+1=4 ways to insert an alphanumeric character into and increase the length of the longest common subsequence by one, we print 4 on a new line.