

Red Sequence

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a sequence of length n .

For each position i , there are between 1 and 3 chess pieces. The chess pieces have three possible colors: red, yellow, and blue.

It is guaranteed that there are no two chess pieces of the same color at the same position.

You need to divide the sequence into several continuous segments.

A continuous segment is called red if the number of red chess pieces in this segment is not less than the number of yellow chess pieces and is also not less than the number of blue chess pieces.

More formally, for a segment:

$$\text{red count} \geq \max(\text{yellow count}, \text{blue count})$$

Your task is to maximize the sum of lengths of all red segments after the division.

Output this maximum possible sum.

Input

The first line contains an integer n ($1 \leq n \leq 1000000$).

The following n lines describe the sequence.

The i -th line contains three integers r_i , y_i , and b_i ($0 \leq r_i, y_i, b_i \leq 1$), where:

$r_i = 1$ means that the i -th position contains a red chess piece, $y_i = 1$ means that it contains a yellow chess piece, and $b_i = 1$ means that it contains a blue chess piece.

It is guaranteed that $1 \leq r_i + y_i + b_i \leq 3$ for every i .

Output

Print one integer: the maximum possible sum of lengths of all red segments.

Example

standard input	standard output
10 1 0 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 1 0 1 0 0 1 1 1 1 1 1 0 1	7