

Problem I

Word Clouds Revisited

Time limit: 2 seconds

When we last visited with Tagg Johnson, he was developing software to create images known as word clouds. Although we will not consider all the details, a word cloud provides a visual representation of textual data in which the size of the bounding box for each word depends on the relative frequency of that word in the original text.

Tagg was surprised to stumble upon an oddity involving his algorithm for laying out words in rows. A maximum width is specified for the size of a word cloud, and Tagg's desire is to place the entries into the cloud in a way that minimizes the overall height. Entries must be placed in a predetermined order, with each subsequent entry either placed horizontally to the right of the previous entry, or else to the far left of a new row. The height of each row is equal to the maximum height of all entries placed in that row, and the overall height of the cloud is equal to the sum of the row heights.

Because his goal is to minimize the height, Tagg's original algorithm would place each entry in the same row as the previous entry, unless it did not physically fit because of the given limit on the width of the cloud. As an example, Figure I.1 shows the layout Tagg's original algorithm produces for a particular cloud with a maximum width of 260 units.

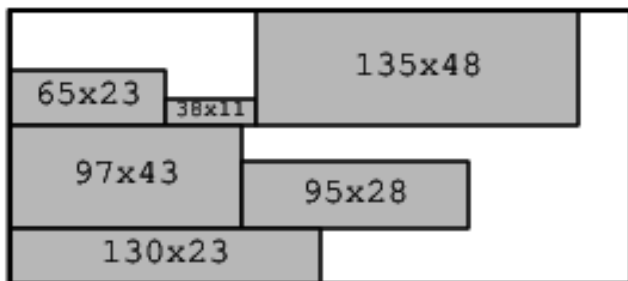


Figure I.1: Tagg's original placement of entries for his word cloud

The first, second, and third entries are placed in the first row (totaling width 238). Then the fourth and fifth entries are placed in a row together (totaling width 193). The sixth entry is by itself on a final row. The overall height of this cloud is $48 + 43 + 23 = 114$ units (as the first row has height 48, the second row has height 43, and the third row has height 23).

However, Tagg later realized that an even better cloud was possible for this same data set, as shown in Figure I.2. By placing the first and second entries in the first row, the third and fourth entries in the second row, and the fifth and sixth entries in a third row, the overall height of the cloud is only $23 + 48 + 28 = 99$ units (while still respecting the overall limit of 260 on the width of the cloud).

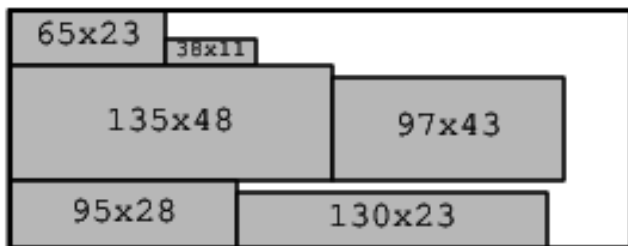


Figure I.2: An optimal placement of the entries from Figure I.1

So now Tagg wants to redesign his software so that, given a list of words and a specific maximum width, it builds a word cloud with the minimum height.

Input

The first line contains two integers, N and C , where $2 \leq N \leq 5000$ is the number of entries to be placed, and $150 \leq C \leq 1000$ is the maximum width for the cloud. Following that are N additional lines, each specifying the bounding box for one entry, in the order in which those entries must be laid out in the cloud. The line describing an entry contains two integers w and h , specifying the respective width and height of the entry, such that $10 \leq w \leq 150$ and $10 \leq h \leq 150$.

Output

Output the minimum height that is required for the word cloud under the restrictions given above.

Sample Input 1

```
6 260
65 23
38 11
135 48
97 43
95 28
130 23
```

Sample Output 1

```
99
```

Sample Input 2

```
3 309
150 100
10 10
150 100
```

Sample Output 2

```
200
```