1. **Sample test 1:**

Hardest Quest:

While playing an RPG game, you were assigned to complete one of the hardest quests in this game.

There are **n** monsters you'll need to defeat in this quest. Each monster **i** is described with two integer numbers - **power**$_i$ and **bonus**$_i$. To defeat this monster, you'll need at least **power**$_i$ experience points. If you try fighting this monster without having enough experience points, you lose immediately. You will also gain **bonus**$_i$ experience points if you defeat this monster. You can defeat monsters in any order.

The quest turned out to be very hard - you try to defeat the monsters but keep losing over and over again. Your friend told you that this quest is impossible to complete. Knowing that, you're interested, what is the maximum possible number of monsters you can defeat?

Input:

he first line contains an integer, n, denoting the number of monsters.
The next line contains an integer, e, denoting your initial experience.
Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, poweri, which represents power of the corresponding monster
Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, bonusi, which represents bonus for defeating the corresponding monster.

Sample Cases:

Input	Output	Output Description
2 123 78 130 10 0	2	Initial experience level is 123 points. Defeat the first monster having power of 78 and bonus of 10. Experience level is now 123+10=133. Defeat the second monster.
3 100 101 100 304 100 1 524	2	Initial experience level is 100 points. Defeat the second monster having power of 100 and bonus of 1. Experience level is now 100+1=101. Defeat the first monster having power of 101 and bonus of 100. Experience level is now 101+100=201. The third monster can't be defeated.

Unique Birthday Gift

Your birthday is coming soon and one of your friends, Alex, is thinking about a gift for you. He knows that you really like integer arrays with interesting properties.

He selected two numbers, **N** and **K** and decided to write down on paper all integer arrays of length **K** (in form **a[1], a[2], ..., a[K]**), where every number **a[i]** is in range from **1** to **N**, and, moreover, **a[i+1]** is divisible by **a[i]** *(where 1 < **i** <= **K**)*, and give you this paper as a birthday present.

Alex is very patient, so he managed to do this. Now you're wondering, how many different arrays are written down on this paper?
Since the answer can be really large, print it **modulo 10000**.

Input:
The first line contains an integer, n, denoting the maximum possible value in the arrays.
The next line contains an integer, k, denoting the length of the arrays.

Sample Cases:

Input	Output	Output Description
2 1	2	The required length is 1, so there are only two possible arrays: [1] and [2].
2 2	3	All possible arrays are [1, 1], [1, 2], [2, 2]. [2, 1] is invalid because 1 is not divisible by 2.
3 2	5	All possible arrays are [1, 1], [1, 2], [1, 3], [2, 2], [3, 3].

2. Sample Test 2:

Bitwise subsequence

You have an array A of N integers $A_1\ A_2\ ..\ A_n$. Find the longest increasing subsequence $A_{i1}\ A_{i2}\ ..\ A_k$ (1 <= k <= N) that satisfies the following condition:

- For every adjacent pair of numbers of the chosen subsequence $A_{i[x]}$ and $A_{i[x+1]}$ (1 < x < k**)**, the expression($A_{i[x]}$ & $A_{i[x+1]}$) * 2 < ($A_{i[x]}$ | $A_{i[x+1]}$) is true

__Note__: '&' is the bitwise AND operation, ' | ' is the bit-wise OR operation

Input:
The first line contains an integer, N, denoting the number of elements in A.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing Ai.

Sample Cases:

Input	Output	Output Description
5 15 6 5 12 1	2	One possible subsequence is: 5 12
6 9 17 2 15 5 2	2	One possible subsequence is: 2 15
7 17 16 12 2 8 17 17	3	One possible subsequence is: 2 8 17

Grid Path

Given a grid. Each cell of the grid contains a single integer value. These values are described by 2D integer array **a** with **N** rows and 2 columns, where **a[i][j]** is the value in the cell located in row **i**, column **j**.

Standing in **(i; j)**, the player can move to any cell of the next row **(i+1; j2)** under the condition that **a[i+1][j2] > a[i][j]**. In other words, the value in the next cell of the player's path should be strictly greater than the value in the current cell of the player's path.
The player can't make any moves if he's already in the last row.

If the player starts in any cell of the first row (either **(1; 1)** or **(1; 2)**), what is the maximum possible total sum of values in all cells he can visit on his path?

Print the answer **modulo 10^9+7**.

Input:
The first line contains an integer, n, denoting the number of rows in a.
The next line contains an integer, 2, denoting the number of columns in a.
Each line i of the n subsequent lines (where $0 \le i < n$) contains 2 space separated integers each describing the row a[i].

Sample Cases:

Input	Output	Output Description
2 1 2 3 4	6	Optimal path is (1;2)->(2;2). The answer is 2+4=6.
2 7 8 5 5	8	No moves are possible from the first row. So start in (1; 2) and collect just 8.
3 1 1 2 2 3 3	6	One of the optimal paths is (1;1)->(2;1)->(3;1). The answer is 1+2+3=6.