Challenge 1

You are currently working with highly specialized devices that may perform a **single background task** and a **single foreground task**. Each task consumes a fixed number of resources **(resource consumption)** from the device’s available **resource capacity**. E.g., a device with a capacity of 3 may run the following configurations:

* A background task that consumes 1 resource and a foreground task that consumes 2 resources or vice versa.
* A background task that consumes 1 resource and a foreground task that consumes 1 resource.

From the previously enumerated configurations, only the former can be considered an **optimal configuration.** A device may not be configured with tasks whose sum of **resource consumption** surpasses the device’s **resource capacity**.

# Challenge

In general, given a set of background tasks and a set of foreground tasks, a device is **optimally configured** when the device is loaded with a background task and a foreground task whose **resource consumption** is equal to or as close as possible to the device’s **resource capacity** without surpassing it.

## Your Task

Given,

* A device with capacity **N**,
* A set of foreground tasks identified by an **ID** and its **resource consumption**
* A set of background tasks identified by an **ID** and its **resource consumption**

Write a program that produces the **ID**s of the combination of tasks that yields an **optimally configured** device. There might be more than one **optimal configuration;** that being the case, produce all of them.

## Sample Input and Outputs

Your program should read a plain text file named *challenge.in* that may contain 1 or more scenarios with the following format:

7

(1, 6), (2, 2), (3, 4)

(1, 2)

The first line represents the **resource capacity** of the scenario’s device. The second line represents a list of the pair (Task **ID**, **resource consumption**) of the scenario’s foreground tasks. The third line represents a list of the pair (Task **ID**, **resource consumption**) of the scenario’s background tasks.

The expected **output** of that scenario should be stored in a plain text file named *challenge.out* as follows:

(3, 1)

That is, the device is **optimally configured** when running the foreground task **ID** 3 and background task 1.

The following is a more comprehensive example of input file along its expected result

|  |  |
| --- | --- |
| ***challenge.in*** | ***challenge.out*** |
| 10  (1, 5), (2, 7), (3, 10), (4, 3)  (1, 5), (2, 4), (3, 3), (4, 2)  20  (1, 9), (2, 15), (3, 8)  (1, 11), (2, 8), (3, 12)  20  (1, 7), (2, 14), (3, 8)  (1, 14), (2, 5), (3, 10) | (1, 1), (2, 3)  (3, 3), (1, 1)  (2, 2) |

**\* Note:** colors above used only to identify the set of inputs that correspond to same-colored output.