Problem 2018.2.1 - Relocation of a company

The furniture and all other objects from one company office space have to be moved at a new location by vans. All items have been packed into boxes that have various sizes (volumes), and their loading into vans is done according to some rules that try to minimize the number of necessary transporter vans. The relocation manager knows that he has at its disposal a sufficient number of vans such that all boxes can be loaded. In each van can be loaded boxes for a total volume of ***D***. Also, the relocation manager knows how many boxes have a given volume. The rules for loading the vans are as follows:

i. the vans are loaded one at the time;,

ii. the van is loaded always with the biggest available box;

iii. the loading passes to the next van ony if in the current van no available, remaining box can be loaded and

iv. all boxes have been loaded into `vans`.

Requirement

Write a program that can help the relocation manager to arrange the boxes into vans in an effective way, according to the mentioned rules.

Input data

One will read from the keyboard (the stdin stream) the following data:

- on the first line: two integer numbers ***D*** and ***k***, separated by space, representing ***D*** – the volume of each van as cubic centimeters, ***k*** – the number of different values of the volume of boxes to be loaded in the vans;
- on the next ***k*** lines: two integer numbers ***n*** and ***p***, on each line, representing the number of boxes ***n*** of volume ***p*** cubic centimeters that have to be loaded in the vans.

The ***k*** lines containing informations about the boxes are given in iverse order of their volume ***p***.
All lines containing input data are ended with the *newline* character (*Enter* key).

Output data

The program will display on screen (the standard output *stream*) ***m*** lines, that correspond to the ***m*** vans that have been loaded with at least one box, in the order of their loading completion (according to the given rules). Each of the ***m*** lines will contain a series of integer numbers, separated by space, representing the volume of the boxes that have been placed in that particular vans, in the order of their placement in the van (according to the given rules).

ATTENTION to the compliance to the problem requirements: the display of results must be done EXACTLY as required! In other words, on the standard output stream there will be nothing displayed in addition to the problem requirements; following the automatic evaluation, any supplemental character displayed, or any display different than the requirements, will produce an eroneous result and will lead to the „Reject" of the solution.

Restrictions and remarks

1. The volume of the vans ***D*** is an integer number in the range [50; 10000].
2. The volume of the boxes ***p*** are integer numbers in the range [1; 1000].
3. The number of boxes of each volume ***n*** are integer numbers in the range [1; 100].
4. It is guaranteed that there are no boxes with volume ***p*** larger than the volume ***D*** of the vans.

5. It is not necessary that the vans should be filled at their maxcimal volume ***D***.
6. **Warning:** According to the chosen programming language, the file containing the code must have one of the extensions .c, .cpp, .java, or .m. The web editor does not add automatically these extensions and the lack of the extensions leads to the impossibility of program compilation!
7. **Warning**: The source file must be named by the candidate as: <name>.<ext> where name is the family name (last name) of the candidate and the extension is the one chosen according to the previous warning. Attention to the restrictions imposed by the Java language regarding the class name and the file name!

Example

Input	**Output**
200 5 2 130 4 120 2 80 3 60 7 50	130 60 130 60 120 80 120 80 120 60 120 50 50 50 50 50 50 50

Explanation

8 transport vans are loade:

Van #1:
- the biggest remaining box (130 cm3),
- then, the biggest remaining box (60 cm3) which still fits inside this van (200 -130 = 70 cm3),
- no other boxes fit inside the remaining space: 70-60 = 10 cm3.

Van #2: identical to van #1.

Van #3:
- the biggest remaining box (120 cm3),
- then, the biggest remaining box (80 cm3) which still fits inside this van (200 -120 = 80 cm3),
- no other boxes fit inside the remaining space – the van is fully loaded.

Van #4: identical to van #3.

Van #5:
- the biggest remaining box (120 cm3),
- then, the biggest remaining box (60 cm3) which still fits inside this van (200 -120 = 80 cm3),
- no other boxes fit inside the remaining space: 80-60 = 20 cm3.

Van #6:
- the biggest remaining box (120 cm3),
- then, the biggest remaining box (50 cm3) which still fits inside this van (200 -120 = 80 cm3),
- no other boxes fit inside the remaining space: 80-50 = 30 cm3.

Van #7:
- the biggest remaining box (50 cm3),
- apoi, în ordine, cele mai mari cutii disponibile care mai încap în această maşină: 50, 50, 50,
- no other boxes fit inside the remaining space – the van is fully loaded.

Van #8:
- the biggest remaining box (50 cm3),
- then, in order, the biggest remaining boxes that still fit inside this van: 50 (last one)

Work time: 120 minutes