任务调度(Schedule)

Description

A HPS cluster is equipped with a unique task scheduler. To be simple, it is assumed that this cluster doesn’t support multiple tasks running at the same time, such that only one task is allowed to be in running state at any moment. Initially, the priority of ever task is denoted by an integer which is called priority number. **The smaller priority number stands for high priority. If two tasks have same task number, the priority is decided in the ASCII order of task name.** Following this policy, resources, such as CPU, are always occupied by the task with minimum priority number. When one task is finished, the one with minimum priority number in the rest tasks is picked to execute. The finished task won’t quit immediately. The priority number is doubled and put back to the task set. Once the priority number is greater or equal to **2^32**, this task is deleted from the task set.

Given initial priority setting of every task, your job is to predict the running order of a batch of tasks.

Input

First line contains two integers, says n and m. n stands for the number of tasks in initial state. m stands for the length of predicted sequence. Every line is ended by a line break symbol. In each one of the following n lines, an integer and a string are included. This string is shorter than 8, which only contains lowercase letters and numbers. The integer is priority number and the string is the task name. The integer and string is separated by space.

Output

At most m lines, each one contains a string. Output the name of tasks according to the order that tasks are executed. If the number of executed tasks is less than m, then output all the executed tasks.

Example

Input

```
3 3
1 hello
2 world
10 test
```

Output

```
hello
hello
world
```

Restrictions

0 <= n <= 4,000,000

0 <= m <= 2,000,000

0 < Initial priority number < 2^32

No tasks have same name

Time: 2 sec

Memory: 512 MB

Hints

Priority queue

描述

某高性能计算集群（HPC cluster）采用的任务调度器与众不同。为简化起见，假定该集群不支持多任务同时执行，故同一时刻只有单个任务处于执行状态。初始状态下，每个任务都由称作优先级数的一个整数指定优先级，**该数值越小优先级越高**；**若优先级数相等，则任务名ASCII字典顺序低者优先**。此后，CPU等资源总是被优先级数最小的任务占用；每一任务计算完毕，再选

取优先级数最小下一任务。不过，这里的任务在计算结束后通常并不立即退出，而是将优先级数加倍（加倍计算所需的时间可以忽略）并继续参与调度；只有在优先级数不小于**2^32**时，才真正退出

你的任务是，根据初始优先级设置，按照上述调度原则，预测一批计算任务的执行序列。

输入

第一行为以空格分隔的两个整数n和m，n为初始时的任务总数，m为所预测的任务执行序列长度，每行末尾有一个换行符

以下n行分别包含一个整数和一个由不超过8个小写字母和数字组成的字符串。前者为任务的初始优先级数，后者为任务名。数字和字符串之间以空格分隔

输出

最多m行，各含一个字符串。按执行次序分别给出执行序列中前m个任务的名称，若执行序列少于m，那么输出调度器的任务处理完毕前的所有任务即可。

样例

见英文题面

限制

$0 \le n \le 4,000,000$

$0 \le m \le 2,000,000$

0 < 每个任务的初始优先级 < 2^32

不会有重名的任务

时间：2 sec

内存：512 MB

提示

优先级队列

UI powered by Twitter Bootstrap (http://getbootstrap.com/).
Tsinghua Online Judge is designed and coded by Li Ruizhe.
For all suggestions and bug reports, contact
oj[at]liruizhe[dot]org.