

G. Boom

time limit per test: 1 second

memory limit per test: 256 megabytes

input: input.txt

output: output.txt

Let's consider the famous game called Boom (aka Hat) with simplified rules.

There are n teams playing the game. Each team has two players. The purpose of the game is to explain the words to the teammate without using any words that contain the same root or that sound similarly.

Player j from team i ($1 \leq i \leq n$, $1 \leq j \leq 2$) is characterized by two numbers: a_{ij} and b_{ij} . The numbers correspondingly represent the skill of explaining and the skill of understanding this particular player has.

Besides, m cards are used for the game. Each card has a word written on it. The card number k ($1 \leq k \leq m$) is characterized by number c_k — the complexity of the word it contains.

Before the game starts the cards are put in a deck and shuffled. Then the teams play in turns like that: the 1-st player of the 1-st team, the 1-st player of the 2-nd team, ... , the 1-st player of the n -th team, the 2-nd player of the 1-st team, ... , the 2-nd player of the n -th team, the 1-st player of the 1-st team and so on.

Each turn continues for t seconds. It goes like that: Initially the time for each turn is t . While the time left to a player is more than 0, a player takes a card from the top of the deck and starts explaining the word it has to his teammate. The time needed for the j -th player of the i -th team to explain the word from the card k to his teammate (the q -th player of the i -th team) equals $\max(1, c_k - (a_{ij} + b_{iq}) - d_{ik})$ (if $j = 1$, then $q = 2$, else $q = 1$). The value d_{ik} is the number of seconds the i -th team has already spent explaining the word k during the previous turns. Initially, all d_{ik} equal 0. If a team manages to guess the word before the end of the turn, then the time given above is subtracted from the duration of the turn, the card containing the guessed word leaves the game, the team wins one point and the game continues. If the team doesn't manage to guess the word, then the card is put at the bottom of the deck, d_{ik} increases on the amount of time of the turn, spent on explaining the word. Thus, when this team gets the very same word, they start explaining it not from the beginning, but from the point where they stopped. The game ends when words from all m cards are guessed correctly.

You are given n teams and a deck of m cards. You should determine for each team, how many points it will have by the end of the game and which words the team will have guessed.

Input

The first line contains two integers n, t ($1 \leq n, t \leq 100$), which correspondingly denote the number of teams and a turn's duration.

Next n lines of the input file contain four integers each: $a_{i1}, b_{i1}, a_{i2}, b_{i2}$ ($1 \leq a_{ij}, b_{ij} \leq 100$) — the skills of the first and the second player of the i -th team. The teams are given in the order in which they play.

The next line of the input file contains integer m ($1 \leq m \leq 100$) the number of cards.

Next $2m$ lines contain the cards' descriptions. Two lines describe each card. The first line of the description contains the word that consists of no more than 20 characters. The words only contain small Latin letters. The second line of the description contains an integer c_k ($1 \leq c_k \leq 100$) — the complexity of the word written on the k -th card. The cards are listed in the order in which they lie in the deck from top to bottom. The words on all cards are different.

Output

Print n lines. On the i -th line first print number s_i the number of points the i -th team wins. Then print s_i space-separated words — the words from the cards guessed by team i in the order in which they were guessed.

Examples

input

2 2
1 1 1 1
1 1 1 1
3
home
1
car
1
brother
1

output

2 home car
1 brother

input

2 4
1 2 2 1
2 3 2 2
4
armchair
3
quetzalcoatl
10
pilotage
5
defibrillator
7

output

2 armchair quetzalcoatl
2 pilotage defibrillator