

HELPER

It's unbelievable, but an exam period has started at the OhWord University. It's even more unbelievable, that Valera got all the tests before the exam period for excellent work during the term. As now he's free, he wants to earn money by solving problems for his groupmates. He's made a *list* of subjects that he can help with. Having spoken with n of his groupmates, Valera found out the following information about them: what subject each of them passes, time of the exam and sum of money that each person is ready to pay for Valera's help.

Having this data, Valera's decided to draw up a timetable, according to which he will solve problems for his groupmates. For sure, Valera can't solve problems round the clock, that's why he's found for himself an optimum order of day and plans to stick to it during the whole exam period. Valera assigned time segments for sleep, breakfast, lunch and dinner. The rest of the time he can work.

Obviously, Valera can help a student with some subject, only if this subject is on the *list*. It so happens, that all the students, to whom Valera spoke, have different, but one-type problems. Valera can solve any problem of subject $list_i$ in t_i minutes.

Moreover, if Valera starts working at some problem, he can break off only for sleep or meals, but he can't start a new problem, not having finished the current one. Having solved the problem, Valera can send it instantly to the corresponding student via the Internet.

If this student's exam hasn't started yet, he can make a crib, use it to pass the exam successfully, and pay Valera the promised sum. Since Valera has little time, he asks you to write a program that finds the order of solving problems, which can bring Valera maximum profit.

Input

The first line contains integers m, n, k ($1 \leq m, n \leq 100, 1 \leq k \leq 30$) — amount of subjects on the *list*, amount of Valera's potential employers and the duration of the exam period in days.

The following m lines contain the names of subjects $list_i$ ($list_i$ is a non-empty string of at most 32 characters, consisting of lower case Latin letters). It's guaranteed that no two subjects are the same.

The $(m+2)$ -th line contains m integers t_i ($1 \leq t_i \leq 1000$) — time in minutes that Valera spends to solve problems of the i -th subject. Then follow four lines, containing time segments for sleep, breakfast, lunch and dinner correspondingly.

Each line is in format H1:M1-H2:M2, where $00 \leq H1, H2 \leq 23, 00 \leq M1, M2 \leq 59$. Time H1:M1 stands for the first minute of some Valera's action, and time H2:M2 stands for the last minute of this action. No two time segments cross. It's guaranteed that Valera goes to bed not before midnight, gets up earlier than he has breakfast, finishes his breakfast before lunch, finishes his lunch before dinner, and finishes his dinner before midnight. All these actions last less than a day, but not less than one minute. Time of the beginning and time of the ending of each action are within one and the same day. But it's possible that Valera has no time for solving problems.

Then follow n lines, each containing the description of students. For each student the following is

known: his exam subject s_i (s_i is a non-empty string of at most 32 characters, consisting of lower case Latin letters), index of the exam day d_i ($1 \leq d_i \leq k$), the exam time $time_i$, and sum of money c_i ($0 \leq c_i \leq 10^6$, c_i — integer) that he's ready to pay for Valera's help. Exam time $time_i$ is in the format HH:MM, where $00 \leq HH \leq 23$, $00 \leq MM \leq 59$. Valera will get money, if he finishes to solve the problem strictly before the corresponding student's exam begins.

Output

In the first line output the maximum profit that Valera can get. The second line should contain number p — amount of problems that Valera is to solve. In the following p lines output the order of solving problems in chronological order in the following format: index of a student, to whom Valera is to help; index of the day, when Valera should start the problem; time, when Valera should start the problem (the first minute of his work); index of the day, when Valera should finish the problem; time, when Valera should finish the problem (the last minute of his work). To understand the output format better, study the sample tests.

Sample tests

Input

```
3 3 4
calculus
algebra
history
58 23 15
00:00-08:15
08:20-08:35
09:30-10:25
19:00-19:45
calculus 1 09:36 100
english 4 21:15 5000
history 1 19:50 50
```

Output

```
150
2
1 1 08:16 1 09:29
3 1 10:26 1 10:40
```

Input

```
2 2 1
matan
codeforces
1 2
00:00-08:00
09:00-09:00
12:00-12:00
18:00-18:00
codeforces 1 08:04 2
matan 1 08:02 1
```

Output

```
3
```

2
2 1 08:01 1 08:01
1 1 08:02 1 08:03

Input

2 2 1
matan
codeforces
2 2
00:00-08:00
09:00-09:00
12:00-12:00
18:00-18:00
codeforces 1 08:04 2
matan 1 08:03 1

Output

2
1
1 1 08:01 1 08:02