

“Argh! I can’t stand it anymore!” exclaimed Dan Hanson, CS Department administrative assistant, as he slammed down the phone.

“What’s the matter, Dan?” asked Valentine McKee, who happened to be in the office making some copies.

“Professors and scheduling. I’m trying to schedule this SSC meeting but as soon as I pick a time and start making calls to confirm it, one of the committee members has a conflict. I feel like this is going to go on forever!” He Said.

“Maybe the computer can help,” suggested Valentine. “This is the computer science department, after all,” she Said.

Input

The first line of input will be the “current” date.

All dates will be specified in the form: *dayname month date*, where:

- *dayname* is single character from the set {M,T,W,R,F} representing Monday, Tuesday, Wednesday, Thursday, and Friday, respectively.
- *month* is an integer such that $1 \leq month \leq 12$.
- *date* is an integer such that $1 \leq date \leq 31$.

The next line of input will contain two integers, *n* and *t*. *n* specifies how many meetings should be scheduled, while *t* indicates how long each meeting will last. *t* will be specified in increments of 15 minutes.

All times will be specified as zero padded 4 digit integers in military time. The first two digits indicate the *hour* (00 corresponds to midnight). The appointment times in the input will be in the range of $09 \leq hour \leq 17$. The last two digits indicate the minute, and will be in the be from the set {00,15,30,45}.

You may assume that *date*, while formally bounded by 31, will not be greater than the number of days in the *month*. You may also assume that all years will *not* be leap years.

The rest of the input will contain the schedules for at most 100 individuals. Each schedule will begin with a line specifying the individual’s name, followed by a list of appointments for that individual, ending with the word ‘done’ on a line by itself. No individual will have more than 100 appointments scheduled. Each appointment will be a date followed by a start time and an end time. No appointment will extend before 9am or after 5pm.

The last line of input will be the word ‘done’ on a line by itself.

No appointment will be before the current date, or more than 1 year after the current date.

Output

Your program should output the first *n* possible meeting times when all the people from the input can meet for the specified amount of time (*t*). Each line of output should contain one meeting date and time. Once a common meeting time is found, it should be considered allocated for all people, and cannot be used to schedule other meetings.

The meetings should be sorted by date and time, with the earliest meeting first.

If less than *n* meeting times are available, print all possible meeting times followed by ‘No more times available’.

Sample Input

```
M 8 21
2 60
Jack Casey
M 8 21 0900 1015
done
Jack Ross
M 8 21 1000 1100
M 8 21 1200 1700
done
Jack Swigert
M 8 21 1600 1700
T 8 22 0900 1000
done
done
```

Sample Output

```
M 8 21 1100
T 8 22 1000
```