

## 480 Tempus Fugit

The owner of a successful small business is so busy that she requires all of her appointments to be scheduled two weeks in advance. She has contacted you to write a program that will help her automate this process. Her work week lasts from 9:00am to 5:00pm, Monday through Friday. She leaves for lunch at 12:30pm and returns at 1:30pm each day.

During the week, she collects appointment requests for two weeks in advance in the order that they were made. Your program will process the requests in the same order, effectively giving priority to those requests that were made the earliest. After running your program, she will confirm the scheduled appointments and reschedule those appointments that the program could not arrange for her.

She asks that your program schedule her appointments in ten-minute blocks that begin on the hour and at ten, twenty, thirty, forty, and fifty minutes past the hour. Any requests that do not begin and/or end at these ten-minute points in time must be scheduled between the ten-minute points which encompass them. For example, an appointment request for 10:15am to 10:30am would be scheduled from 10:10am to 10:30am.

She also wants your program to insert a ten minute break after each scheduled appointment so that no two appointments occur in immediate succession. The ten-minute break will occupy the next ten-minute block in her schedule. Ten-minute breaks are not to be added for appointments which can be scheduled to end at 12:30pm or 5:00pm. No appointment is permitted to extend into the lunch period, or to be scheduled to begin during the lunch period. She would like no more than four hours of appointments scheduled on any one day. When computing the four-hour limit, use the adjusted length of the appointment (do not use the original request and do not include the ten-minute breaks which follow the adjusted appointment times). Thus, the appointment request for 10:15am to 10:30am contributes 20 minutes to the four-hour time limit for that day.

Finally, the owner tells you that if an appointment cannot be scheduled as requested, your program should attempt to reschedule the appointment at the same time on each successive day of the week, scheduling it on the first day that it fits: if an appointment cannot be scheduled at 10:00am on Wednesday, your program should try to schedule it on Thursday at 10:00am, then at Friday at 10:00am if it cannot be scheduled on Thursday (do not attempt to schedule the appointment earlier in the week or at different times of the day). If your program cannot schedule the appointment in this manner, the name of the person requesting the appointment should be added to the list of appointments that could not be scheduled.

### Input

The input file will consist of at most 25 appointment requests on a separate line in the following format:

*name day start\_time duration*

where the name field occupies the first ten character positions of the request, the day field occupies the next three character positions, and the remaining two fields each consist of a pair of integers. The name field will contain ten alphanumeric characters and the day field will contain one of the three character sequences: MON, TUE, WED, THU, or FRI. Following the day will be two times: the starting time of an appointment request, and its duration. Both times consist of two integers, hours followed by minutes. For example, the appoint request:

Johnstone TUE 09 15 1 30

means that Johnstone requests an appointment for the Tuesday of that week starting at 9:15am for one hour and thirty minutes. Since the schedule is in ten-minute blocks, your program would attempt to

schedule the appointment for Tuesday from 9:10am to 10:50am (adding 1 hour and 40 minutes to the total appointment time on Tuesday).

Your program may assume that all fields in an appointment request will contain legitimate data and that times will be expressed in terms of the maximal number of hours (9:15am will always be 9 15 rather than 8 75 and ninety minutes will always be 1 30 rather than 0 90).

## Output

Your output will consist of a schedule for the week followed by a summary listing the names of those people whose appointments could not be scheduled during this time. Days with no scheduled appointments should be noted.

## Sample Input

```
Johnstone TUE 09 15 1 30
Peterson MON 09 00 0 30
McKeever FRI 09 30 1 00
Garzarelli THU 10 45 0 20
Tucker MON 10 00 2 30
Davis MON 02 30 1 00
Corrigan MON 02 00 0 15
Trump WED 01 00 3 00
Logan THU 09 45 1 05
Schulman THU 11 10 0 30
```

## Sample Output

```
APPOINTMENT SCHEDULE FOR THE WEEK
MONDAY
Peterson 9:00 to 9:30
Tucker 10:00 to 12:30
Davis 2:30 to 3:30
TUESDAY
Johnstone 9:10 to 10:50
Corrigan 2:00 to 2:20
WEDNESDAY
No Appointments Scheduled
THURSDAY
Garzarelli 10:40 to 11:10
FRIDAY
McKeever 9:30 to 10:30
Schulman 11:10 to 11:40
APPOINTMENTS COULD NOT BE SCHEDULED FOR:
Trump
Logan
```