2013 Fall Computer Science I Program #3: Organ Transplant Database
Please consult Webcourse for the due date/time

Note: For this assignment, you will be writing a program that implements a different solution to the problem posed in the first program for the course. The only differences in this assignment compared to program #1 will be in the section labeled "**Specification Details**", **one tiny change in organT, and the total number of donated organs received in the input file.** Since most of the sample cases are already known, the grading criteria for this assignment will be posted beforehand. For convenience, the whole description of the problem is reposted below.

The United States as an Organ Transplant database, to keep track of those who need organ transplants. Depending on a number of factors, when matches are found, organs are donated to those on the list. Typically, one can only receive a donation if the donor's organ is a match for the recipient. In this program, we will simulate a simplified organ transplant data base. To simplify the problem, our data base will simply keep track of the following information for each person in need of an organ:

1) Person's name
2) The organ they need to replace.
3) Their blood type (A+, A-, B+. B-, AB+, AB-, O+, O-)
4) The date (month, day, year) they were added to the list.
5) The time (hours, minutes) in military time they were added to the list.

Your program should use the following structs and constants:

```
#define SIZE 20
#define BLOODTYPESIZE 4

typedef struct {
    int month;
    int day;
    int year;
} dateT;

typedef struct {
    int hour;
    int minute;
} timeT;

typedef struct {
    char name[SIZE];
    char organname[SIZE];
    char bloodtype[BLOODTYPESIZE];
    dateT dateAdded;
    timeT timeAdded;
} organT;
```

The name component will store the patient's name, organname will store the name of the organ they are in need of, bloodtype will store one of the 8 aforementioned strings representing bloodtype, the

dateAdded and timeAdded will store when the corresponding organ was put on the waitlist while received will store a 0 initially and may store a 1 to indicate that the organ has been transplanted. (Note: In this assignment you will not physically remove the record of any patient, even if they have received their organ. This field will be solely used so that you can skip matching an organ to a patient who has already received one.)

Your program should read in information for a file consisting of the organ wait list, as well as a sequence of organs received for donation. For each organ received, your program should print out the name of the person and the organ they have received. The organ should go to the the person who has been on the waitlist the longest, who is a match for the organ. For the purposes of this assignment, a match occurs when the donor organ is the same AND the bloodtype of the donor is the same as the recipient. Once a match is found for an organ, they should not be matched again.

Input File Format

The first line of the input file will contain a single positive integer, n $(n \leq 120000)$, representing the number of organs on the waiting list. The next n lines will contain information about one organ each. Each of these lines will contain the person's name, the organ they need replaced, their blood type, the date they were added to the organ database and the time they were added to the organ database. Each of these items will be separated by a space. All names will be comprised of letters and underscores only, all organ names will be comparised of lowercase letters, all bloodtypes will be one of the previously mentioned 8 strings, all dates will be of the format $m/d/y$, where $m, d,$ and $y,$ represent the numeric month day and year the patient was added to the organ donation list (for this particular organ). Finally, the time will be of the form $hr:min$, where $hr(0 \leq hr \leq 23)$ and $min(0 \leq min \leq 59)$ represent the numeric hour and minutes for the time the patient was added to the organ donation list. You are guaranteed that no two organs were added to the list on the same date and time and that no name or organ name will contain more than 19 characters.

The following line of the input file (line number $n+2$), will contain a single positive integer, k $(k \leq 100000)$ representing the number of organs received, during some fixed period in time. The following k lines will contain information about the organs received, in the order they were received. Each of these lines will contain two strings separated by a space: the name of the organ and the blood type of the donor. These will both adhere to the specifications previously given.

Output Specification

Output a single line for each organ received. If a matching recipient exists in the database that hasn't yet received an organ, print out the name of the recipient, followed by the organ they received. If no match exists in the database, print out the following on a single line.

```
No match found
```

Input and Output

I/O for this assignment will be the same five files as assignment #1 as well as one new file with more organs received.

Specification Details

You must store each organ request in a node of a binary search tree. When reading in the list of patients waiting for an organ, you must insert each item into the binary search tree. When processing organs received, you must find the appropriate node in the tree that corresponds to the patient to receive the organ, output his/her name, and delete this node from the tree.

In order to implement this assignment using a binary tree, you must have a method for comparing two organs. **Please use the following:**

For both organs, create a concatenated string, organ name plus blood type. For example, using organ0.in, this concatenated string for John_Lynch is "kidneyA-".

Compare both organs using their concatenated strings.

If this doesn't break the tie, then do so using the dates the two patients have been waiting from (earlier date coming first), and if this is a tie, break it using the time. You are guaranteed that no two patients will have the same organ type, blood type, date and time in the input file.

Deliverables

Please turn in a single source file, *organbintree.c*, with your solution to this problem via Webcourses before the due date/time for the assignment. Make sure that your program reads form standard in and outputs to standard out, as previously shown in lab.