1. Problem 1075

Title: Footbal Statistics I

Description

In this assignment, you are asked to process a list of outcomes of football games and process the results.

You can assume that there are only 30 teams, but not much else:

- not every teams need to play against every other team
- some teams might not play a single match
- the number of matches between two teams need not be the same as that between other teams
- the number of home games might be different from the number of away games
- the total number of match results that you may need to process can be small, or *very large*.

The last point means that you cannot store the entire list of match results in memory, only aggregate statistics.

Since the statistics are fractional, make sure that you process the data using double precision.

Input

The input consists of:

- The number of matches played
- List of results for each match

Each row in the list has the form

```
home_team_id away_team_id goals_home_team goals_away_team
```

where the IDs are integers in the range [0, 29]. That is, you can assume that there are maximally 30 teams.

For instance:

```
3 8 1 0
```

encodes a 1-nil win for home team 3 against away team 8, and

```
16 1 0 0
```

encodes a draw between teams 16 and 1.

Output

For each team that played at least one match, output a row containing:

1. team id
2. win ratio (= #wins / #plays)
3. win ratio on home games. This should be -1 in case of no home games.
4. average point difference in won games. This should be -1 in case of no won games.
5. the largest point difference in a single game.
6. the ID of the opponent against which the largest point difference occurred.
7. the largest **average** point difference, over multiple games, against one particular opponent,
8. as well as the ID of that opponent.

All fractional output data should be truncated to 3 digits behind the decimal point.

Sample Input (1)

```
2
0 1 5 0
2 3 0 5
```

Sample Output (1)

```
0 1.000 1.000 5.000 5 1 5.000 1
1 0.000 -1 -1 -5 0 -5.000 0
2 0.000 0.000 -1 -5 3 -5.000 3
3 1.000 -1 5.000 5 2 5.000 2
```

Sample Input (2)

```
5
3 5 4 2
3 1 0 0
2 7 1 0
4 3 3 0
2 3 0 2
```

Sample Output (2)

```
1 0.000 -1 -1 0 3 0.000 3
2 0.500 0.500 1.000 1 7 1.000 7
3 0.500 0.500 2.000 2 5 2.000 2
4 1.000 1.000 3.000 3 3 3.000 3
5 0.000 -1 -1 -2 3 -2.000 3
7 0.000 -1 -1 -1 2 -1.000 2
```

Sample Input (3)

```
4
0 1 5 0
2 3 0 5
20 21 3 0
21 20 4 0
```

Sample Output (3)

```
0 1.000 1.000 5.000 5 1 5.000 1
1 0.000 -1 -1 -5 0 -5.000 0
2 0.000 0.000 -1 -5 3 -5.000 3
3 1.000 -1 5.000 5 2 5.000 2
20 0.500 1.000 3.000 3 21 -0.500 21
21 0.500 1.000 4.000 4 20 0.500 20
```

1. Problem 1076

Title: Matrix Multiplication of Sparse Matrices

Description:

In this exercise you have to write a program that can compute the matrix product of two **sparse** matrices that are represented as lists in the input format. (i.e., the matrices have 0 for all non-specified entries, for more information see the end of this assignment).

Make sure that you do not use more memory than needed to store the input.

Input:

The first number of the input, N, describes the number of entries of the first matrix A. Then the entries of A follow (see below), then a number, M, which indicates the number of entries of the second matrix B. Then the actual entries of B will follow.

The entries of the matrices A, B, are encoded, 1 per line, in the following format:

```
row_id column_id value
```

You can assume that entries are sorted first by row, then by column. The `values` themselves are integers, but may be negative (e.g., see the "-1" entry in the sample input)

Output:

A sparse matrix representation of the output matrix. Entries are sorted first by row, then by column.

Sample Input:

```
3
0 0 4
0 1 2
1 0 -1
2
1 0 10
1 1 -10
```

Sample Output:

```
0 0 20
0 1 -20
```

About Sparse Matrix Representations

A 'sparse' representation of a matrix, is one that only encodes the non-zero entries:
https://en.wikipedia.org/wiki/Sparse_matrix

The encoding that this assignment uses is the 'coordinate list' encoding, with the additional condition of ordering (i.e., "You can assume that entries are sorted first by row, then by column.").

The following bit of octave code illustrates what is going on:

```
$ octave
GNU Octave, version 3.8.2
Copyright (C) 2014 John W. Eaton and others.
This is free software; see the source code for copying conditions.
<...SNIP...>
Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

                                                                    #the first matrix:
octave:1> A = [0,0,4; 0,1,2; 1,0,-1]
A =
   0   0   4
   0   1   2
   1   0  -1

                                                  #because octave does start indexing from '1':
octave:2> A=[ A(:,1)+1, A(:,2)+1. A(:,3) ]
A =
   1   1   4
   1   2   2
   2   1  -1

                                                              #interpret A as a sparse matrix:
octave:3> As = spconvert(A)
As =
Compressed Column Sparse (rows = 2, cols = 2, nnz = 3 [75%])

  (1, 1) ->  4
  (2, 1) -> -1
  (1, 2) ->  2

                                          #get the traditional 'dense' representation of A:
octave:4> Adense = full(As)
Adense =
   4   2
  -1   0



                                                                    #the second matrix:
octave:5> B = [1,0,10; 1,1,-10]
B =
    1    0   10
    1    1  -10

                                                  #because octave does start indexing from '1':
octave:6> B=[ B(:,1)+1, B(:,2)+1. B(:,3) ]
B =
    2    1   10
    2    2  -10

                                                              #interpret as a sparse matrix:
octave:7> Bs = spconvert(B)
Bs =

Compressed Column Sparse (rows = 2, cols = 2, nnz = 2 [50%])

  (2, 1) ->  10
  (2, 2) -> -10

                                                   #get the traditional 'dense' representation:
octave:8> Bdense = full(Bs)
Bdense =
    0    0
   10  -10

octave:9> C = Adense *Bdense
C =
   20  -20
    0    0

octave:10>
```

1. Problem 1077

Title: Birthday Lookup

Description

In this assignment, you will need to create a 'database' of birthdays that will be queried repeatedly.

First, a (non-specified) number of birthdays + names is specified. You will need to store this data in a binary search tree (BST). The nodes of the BST will contain the (birthday, name) pairs.

Once the birthdays are stored, a number of queries follows. These take the shape of numeric (D-M-YY) dates. For each queried date you need to return the person that has a birthday on that date, or, if the queried date is nobodies birthday, the name of the person whose birthday is the first after the query date.

- Each line corresponds to a **valid** date, and a name. (You do not have to worry about date validation; all dates in the input are valid dates.)
- Each date consisting of one string ("January", "February", ..., or "December"), one integer between 1 and 31, and one two digit integer representing the year (from 90 to 99, and then from 00 to 16).
- Each name consists of a first name and a last name. (i.e., you do not need to worry about compound names comprised of more than 2 'words')
- Please use structures to store the dates and names.
- Use malloc to dynamically allocate just enough space for all the data and data structures.
- The access time of binary trees depends on how "balanced" they are. So a perfect solution would deal with this issue.

Input

- the string "BIRTHDAYS_START"
- followed by the (date, birthday)-pairs
- followed by the string "QUERIES_START"
- followed by a number of user query in format *day month year* (e.g. "1 1 00" or "31 3 68").

Output

- the names of persons who have their birthday on the queried dates, or those of the first subsequent birthdays.
- In the latter case, this should be indicated by pre-pending "first subsequent birthday: " before the name.
- In the unlikely case that there is no next birthday, you should print "no subsequent birthday"

Sample Input

```
BIRTHDAYS_START
January 1 01 Molly Mcauliffe
January 1 00 Dennise Nigh
February 28 99 Erma Merrick
July 17 12 Linn Alvin
September 10 12 Delphia Bynum
July 1 00 Vania Jones
June 30 90 Brittney Gemmill
August 25 06 Aubrey Sherard
May 27 08 Marica Rising
October 1 03 Eugena Steele
QUERIES_START
1 1 00
3 7 12
30 6 90
6 7 16
1 1 90
```

Sample Output

```
Dennise Nigh
first subsequent birthday: Linn Alvin
Brittney Gemmill
no subsequent birthday
first subsequent birthday: Brittney Gemmill
```