

COP 3223 Assignment #8

Program A: Greeting (greeting.c):

Your friend is writing a program that will be used by many different types of people. He's so busy focusing on making it perfect for everyone that he has decided to recruit some friends to help him complete some of the smaller, but certainly still important, parts. He's asked you to write the greeting interface.

Given the user's name and their name prefix, output a welcome message to them. A name will always be a string that is less than 100 characters in length and does not contain spaces. A prefix will be input as an integer to select from a list. See the sample input for the list that should be used. If an option is input that is not on the list, just leave off the prefix.

You may write any functions you find useful in this program, but you are not required to have any for this program.

Sample Run (user input in bold and italics):

What is the user's name?

Boswell

What is the user's name prefix?

- (1) Mr.
- (2) Ms.
- (3) Mrs.
- (4) Dr.

1

Hello Mr. Boswell, welcome to the program!

Sample Run (user input in bold and italics):

What is the user's name?

John

What is the user's name prefix?

- (1) Mr.
- (2) Ms.
- (3) Mrs.
- (4) Dr.

0

Hello John, welcome to the program!

Program B: Matrix Multiply (matrixmultiply.c):

In mathematics, especially in the context of programming, it is quite common to work with matrices. A matrix (plural matrices) is a 2D grid of numbers used to represent data in various ways. There are many different interesting ways to solve all kinds of challenging problems using matrices. But let's just stick to something basic for now. Often it is useful to multiply two matrices together, resulting in a single result matrix. Given two matrices, multiply them and output this result.

Normally matrices have varying height and width, and there are specific restrictions on what each of these can be for each matrix in order to multiply them together. For simplicity, though, we'll just say both of the matrices are NxN in size, which results in a result of size NxN.

In order to compute a cell (row, column) of a result matrix, R, of the multiplication of A and B, you can use the following formula (where each matrix is NxN):

$$R[r][c] = \sum_{i=0}^N A[r][i] * B[i][c]$$

In other words, each cell of the result matrix is the (dot) product using respective pairs of the corresponding cells in the row of A and column of B. If you still find this unclear, you may choose to consult the many Google results you will get for "matrix multiplication" (of two matrices), since as stated, this is a very common procedure in mathematics and as such there are many different ways to explain it.

The first line of input will be a positive integer N. The following lines will be the NxN numbers in A, followed by then NxN numbers in B. You should then output the product A*B, as a grid (matrix).

You may write any functions you find useful in this program, but you are not required to have any for this program.

Sample Run (User input in bold and italics):

```
3
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
16 17 18
84 90 96
201 216 231
318 342 366
```

(Note: spacing in the above sample was modified for clarity. If you want to do this, you can use "%3d" instead of "%d" to output your numbers. But this is not required.)

Sample Run (User input in bold and italics):

```
3
5 3 2
1 5 2
6 3 6
2 5 8
1 3 2
5 2 1
23 38 48
17 24 20
45 51 60
```

Sample Run (User input in bold and italics):

```
2
1 1
1 0
1 1
1 0
2 1
1 1
```

Program C: Birthday Lookup (birthdays.c):

Recently, you've found yourself having difficulty remembering all of your many friends' birthdays. So you've decided to write a program to help you (because really, who would want to rely on silly social networking sites' reminders when you can write your own program to do it?).

Write a program that will read a list of names and birthdays from the file "birthdays.txt" into your program. The format of this file will first be an integer, N ($1 \leq N \leq 50$), followed by N lines that contain the first name, last name, and birthday of a person, separated by a spaces. The program should then read a series of birthday queries from standard input. A birthday query should always consist of exactly 5 characters of the form "MM/DD", where MM is the two digit month and DD is the two digit day. Birthdays in the file will correspond to this format as well. Names will each consist of a first and last name, separated by a space. First and last names will not contain any spaces and will each be at most 50 characters. For each query, in order, output the names of everyone who has a birthday on that day. The queries will end with the word "DONE".

Hint: because the date format is the same in the file and the queries, your program doesn't need to understand the format, it should just look for an exact string match.

Sample Run (User input in bold and italics):

Enter a date (MM/DD) or "DONE":

03/29

The following people have birthdays on 03/29:

John Tyler

Enter a date (MM/DD) or "DONE":

08/04

The following people have birthdays on 08/04:

Barack Obama

Enter a date (MM/DD) or "DONE":

11/02

The following people have birthdays on 11/02:

James Polk

Warren Harding

Enter a date (MM/DD) or "DONE":

01/01

The following people have birthdays on 01/01:

(None)

Enter a date (MM/DD) or "DONE":

DONE

The input for the above sample was using a file containing the birthdays of all US presidents. See the assignments section under Files on webcourses for this sample file ("birthdays.txt").

Deliverables:

Please submit three separate .c files for your solutions to these problems via WebCourses by the designated due date:

Program A: **greeting.c**

Program B: **matrixmultiply.c**

Program C: **birthdays.c**