CIS*1300: ASSIGNMENT 3 – cScheduler

/**!**\ **Note** that there will be no FACE interviews for A3 due to time constraints. Weight of A3 code submission is 9%.

Concepts required:
Text files, Command-line arguments, 1 and 2D arrays, 1 and 2D Strings, Functions, and concepts learnt in developing assignments 1 and 2

1.0 Contextual Situation: Course Assignment

Congratulations! You have been hired as a developer for the Computer Science to help the department with course assignment for the current year. As a developer for this software (cScheduler), you have been selected to complete the following tasks below to be used within the system.

There are currently 6 professors and 10 courses that are offered this year. The system tracks various info such as the number of courses taught by a Professor this year and other info that is detailed in the next section. Furthermore, the system also generates a report as shown in task 9 (as an additional functionality task).

1.1 Prerequisite – First things First

You are required to download the given header file called **givenA3.h**, a text file called **courses.txt** and a text file called **data.txt**. You will need to enter file names for tasks 1 and 2 as command-line arguments – read Zybooks 2.24 and 13.7 for more on command-line arguments.

You will need to use the following constants defined in givenA3.h:

```
#define NUMBER_PROFS 6
```
This represents the number of professors.

```
#define NUMBER_COURSES 10
```
This represents the number of courses.

2.0 Your Task

Part A: You must write a **SEPARATE** user-defined function definition, for each and every one of the following tasks**. You must do this in a file called lastnameFirstnameA3.c.**

Task 1: Data entry of course IDs, course names and professor names

**** Function prototype (also given in givenA3.h) ****

CIS*1300: ASSIGNMENT 3 – cScheduler

```
void readCourseProfData (
     char filename [50],
     char courseNames [NUMBER_COURSES][50],
     char profNames [NUMBER_PROFS][30],
     int courseID [NUMBER_COURSES]
);
```

In this task, you have to read courseNames, profNames and courseID from a text file called courses.txt to populate 3 different arrays courseNames, profNames and coursed. The text file name “courses.txt” must be read as the first command-line argument (e.g. ./a.out courses.txt data.txt) and not from standard input. Note that the first 10 lines of this file store 10 course names, the next 6 lines store 6 professor names and the following 10 lines store 10 course IDs. Also note that course names and professor names may have multiple words in them (e.g. Introduction to Programming). All course IDs are integers.

Task 2: Data entry of course assignments of each professor

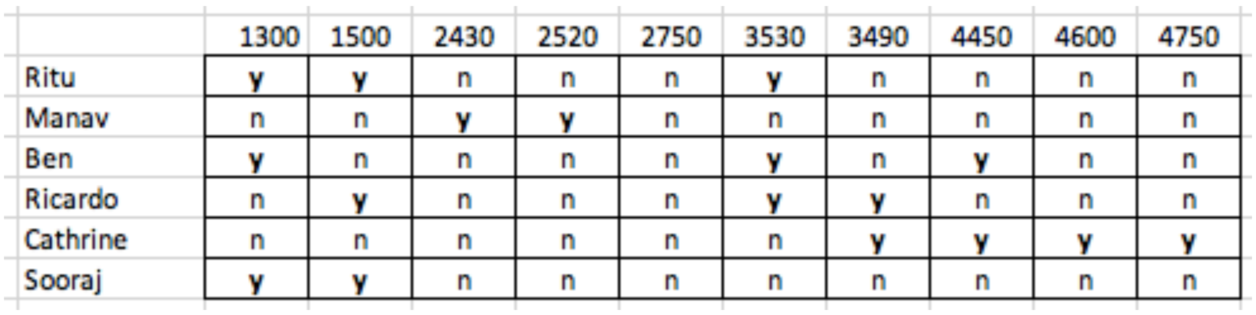
**** Function prototype (also given in givenA3.h) ****

```
void readData (
     char filename [50],
     char data [NUMBER_PROFS][NUMBER_COURSES]
);
```

/!\ Note: Uppercase and lowercase letters are both acceptable. Therefore, your function should accept both “Y” and “y” for yes and “N” and “n” for no. It would be beneficial to store the lowercase version within the array for simplicity later.

/!\ Hint: The ctype.h library has some helpful functions such as tolower ().

These course assignments **MUST** be read from a text file called **data.txt** and stored in the 2-dimensional array called **data** that is accessible outside of the function. The text file name “data.txt” must be read as the second command-line argument (e.g. ./a.out courses.txt data.txt) and not from standard input. An example 2D array created in this function is shown below:



	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

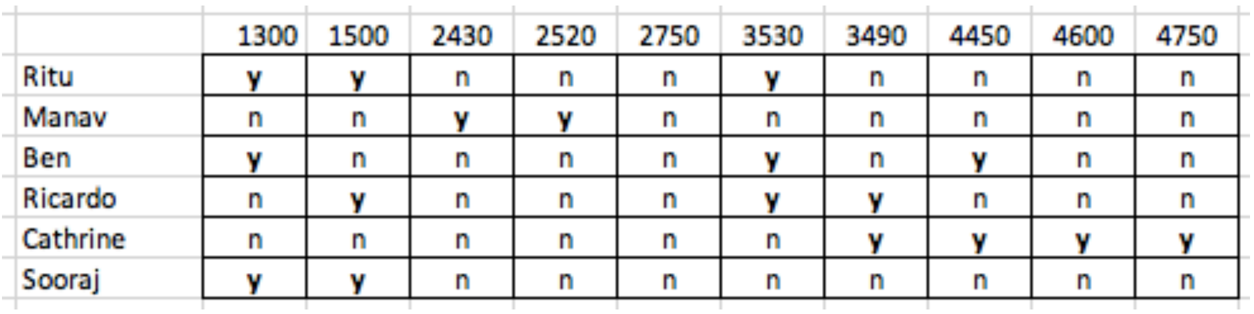
CIS*1300: ASSIGNMENT 3 – cScheduler

Task 3: Profs who teach n or more courses (n >= 0)

**** Function prototype (also given in givenA3.h) ****

```
int numProfsTeachingNCourses (
      char data [NUMBER_PROFS][NUMBER_COURSES],
      int n,
      char profNames [NUMBER_PROFS][30]
);
```

This function must print the names of all professors who teach n or more courses. It then returns the total number of professors who teach n or more courses.
For example, if data has the following course assignment and n = 3, then the function must print “Ritu”, “Ben”, “Ricardo” and “Cathrine” and return 4.



	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

Task 4: Profs who teach only nth-level courses (and no other course) (n >=1 and n <=4)

**** Function prototype (also given in givenA3.h) ****

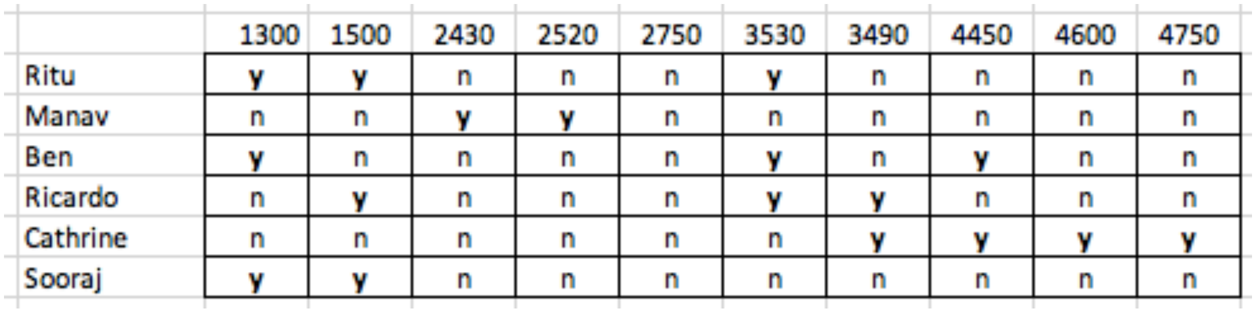
```
int numProfsTeachingNLevelCourses(
      char data[NUMBER_PROFS][NUMBER_COURSES],
      int n,
      int courseID[NUMBER_COURSES],
      char profNames[NUMBER_PROFS][30]
);
```

This function must print the names of all profs who teach only nth level courses. It also returns the total number of profs teaching only n-th level courses.

In the table below, for example, the prof who only teaches 100-level course (and no other course) is “Sooraj”, whereas the prof who teaches only 200-level courses (and no other course) is “Manav”.

CIS*1300: ASSIGNMENT 3 – cScheduler

/!\ Hint: Use the % or / operator on courseID to extract its first digit and use it to decide if it is a 100-level, (meaning, columns 0 and 1), 200-level course (meaning, columns 2, 3 and 4) and so on.



	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

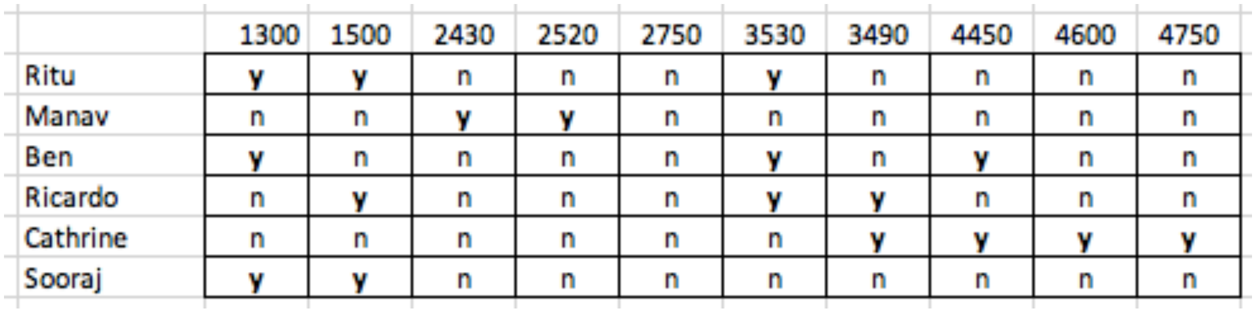
Task 5: Courses that have n Profs assigned this year (n >= 0)

**** Function prototype (also given in givenA3.h) ****

```
int coursesWithNProfs (
      char data[NUMBER_PROFS][NUMBER_COURSES],
      int n,
      char courseNames[NUMBER_COURSES][50]
);
```

This function returns the total number of courses that are taught by n professors. It also prints the name of all courses that are taught by n professors.

For example, if the course assignment is as shown in the table below and n = 3, then it prints course names “Programming”, “Introductory Programming” and “Intro to Databases”.



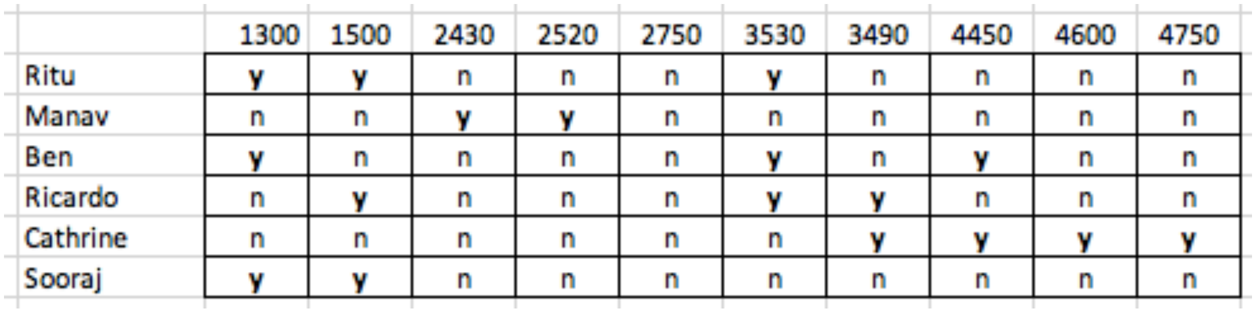
	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

Task 6: Find the average number of courses taught by a prof.

**** Function prototype (also given in givenA3.h) ****

```
float avgNumCourses (char data[NUMBER_PROFS][NUMBER_COURSES]);
```

This function returns the average number of courses taught by a professor. ~~Note that since the average is typically a float value, you must round it to the next integer when you return the value.~~



	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

~~For the above example, the average number of courses taught by a Prof is 17 / 6 = 2.83, therefore the function must return 3.~~

<u>Task 7: find course Name, given course ID</u>

**** Function prototype (also given in givenA3.h) ****

```
int getCourseName (
      int courseNum,
     int courseID[NUMBER_COURSES],
     char cNameFound [50],
     char courseNames[NUMBER_COURSES][50]
)
```

This function, given a course number, searches for its course name and stores it in a string parameter (e.g. cNameFound). It returns 1 if the course is found, 0 otherwise. For example, if the course number is 1300, it will return 1 and store “Programming” in cNameFound.

<u>Task 8: find course num, given course name</u>

**** Function prototype (also given in givenA3.h) ****

```
int getCourseNum (
     char cName [50],
     int courseID[NUMBER_COURSES],
     int * cNumFound,
     char courseNames[NUMBER_COURSES][50]
);
```

This function, given a course name, searches for its course number and stores it in an output int parameter (e.g. cNumFound). It returns 1 if the course is found, 0 otherwise. For example, if the course number is 1300, it will return 1 and store 1300 in *cumFound. Remember to use fgets since some course names have multiple words. Also that fgets includes the \n character!

Part B: You must write a main program that displays a menu of tasks 3, 4, 5, 6, 7 and 8, prompts the user to enter a choice and performs the appropriate task based on the user's choice. If you do the additional task (task 9 below), then your menu must display task 9 as well. Your program must end if the choice entered is any number other than 3, 4, 5, 6, 7, 8 or 9. **Call this program lastnameFirstnameA3Main.c.**

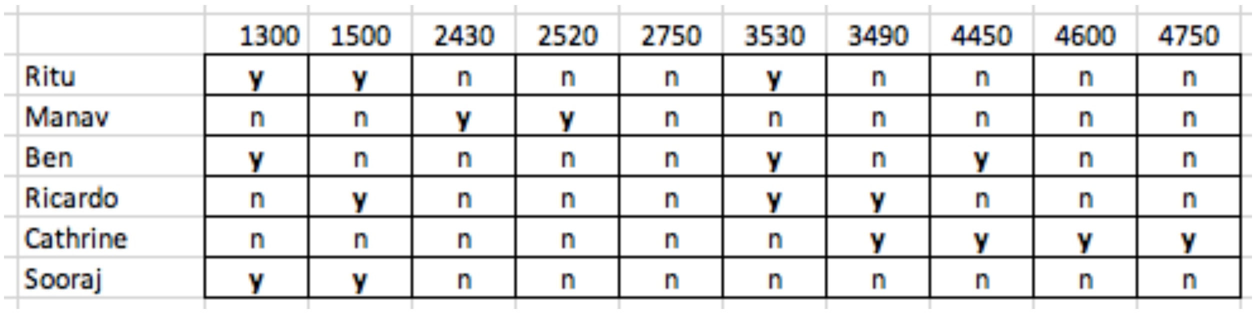
3.0 Extras – additional functionality

By completely the above task correctly, you have the potential to earn up to 90% of the grade for this assignment. By going beyond the material given and completely the extra section (task 9), you have the potential of earning another **10%.**

Task 9: Generate a report as shown below

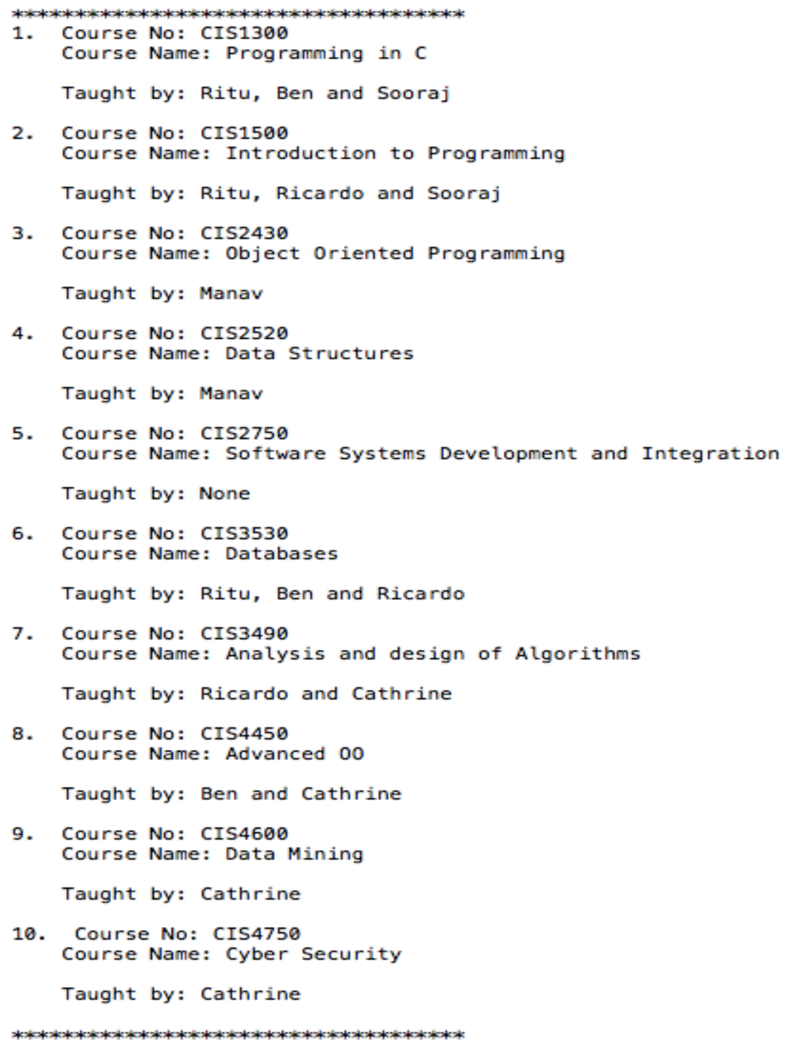
**** Function prototype (also given in givenA3.h) ****

```
void generateReport(
     char data[NUMBER_PROFS][NUMBER_COURSES],
     int courseID[NUMBER_COURSES],
     char courseNames[NUMBER_COURSES][50],
     char profNames[NUMBER_PROFS][30]
)
```



	1300	1500	2430	2520	2750	3530	3490	4450	4600	4750
Ritu	y	y	n	n	n	y	n	n	n	n
Manav	n	n	y	y	n	n	n	n	n	n
Ben	y	n	n	n	n	y	n	y	n	n
Ricardo	n	y	n	n	n	y	y	n	n	n
Cathrine	n	n	n	n	n	n	y	y	y	y
Sooraj	y	y	n	n	n	n	n	n	n	n

Output produced by task 9 for the given data is as follows:



```
***********************************
1.  Course No: CIS1300
    Course Name: Programming in C

    Taught by: Ritu, Ben and Sooraj

2.  Course No: CIS1500
    Course Name: Introduction to Programming

    Taught by: Ritu, Ricardo and Sooraj

3.  Course No: CIS2430
    Course Name: Object Oriented Programming

    Taught by: Manav

4.  Course No: CIS2520
    Course Name: Data Structures

    Taught by: Manav

5.  Course No: CIS2750
    Course Name: Software Systems Development and Integration

    Taught by: None

6.  Course No: CIS3530
    Course Name: Databases

    Taught by: Ritu, Ben and Ricardo

7.  Course No: CIS3490
    Course Name: Analysis and design of Algorithms

    Taught by: Ricardo and Cathrine

8.  Course No: CIS4450
    Course Name: Advanced OO

    Taught by: Ben and Cathrine

9.  Course No: CIS4600
    Course Name: Data Mining

    Taught by: Cathrine

10.  Course No: CIS4750
    Course Name: Cyber Security

    Taught by: Cathrine
***********************************
```

3.0 Program Submission and Administration Information

The following section outlines what is expected when submitting the assignment and various other administration information with respect to the assignment.

3.1 Program Submission

You must submit 2 files **lastnameFirstnameA3.c** and **lastnameFirstnameA3Main.c**
Incorrect file names will result in penalty.

3.2 Program Expectations

Your program is expected to follow the outlined information exactly. Failure to do so will result in deductions to your assignment grade.

- The program files you submit MUST compile with NO warnings and errors using the flags - std=c99 and -Wall.

- If you decide to define and use helper functions (other than what is listed in the .h file), then you must also add their prototypes in lastnameFirstnameA3.c.
- **Programs that fail to compile will be given a mark of 0.**
- Programs that produce warnings upon compilation will receive a deduction of 1 mark for each type/category of warning.
- The program file must contain the header comment shown in section Program Header Comment Format with the **underlined & bolded** sections properly filled in.

3.3 List of "Do Not"s

- Do not use Global Variables
  - Using global variables will result in 0 in the assignment
- Do not use GOTO statements
  - Using Goto statements will result in 0 in the assignment

3.3 Program Header Comment Format

```
/***********************lastnameFirstnameA3.c**************
  Student Name: Ritu Chaturvedi       Email Id: ritu
  Due Date:                    Course Name: CIS 1300
  I have exclusive control over this submission via my password.
  By including this statement in this header comment, I certify that:
   1) I have read and understood the University policy on academic
      integrity;
   2) I have completed the Computing with Integrity Tutorial on
      Moodle; and
   3) I have achieved at least 80% in the Computing with Integrity
       Self Test.
    I assert that this work is my own. I have appropriately acknowledged
    any and all material that I have used, whether directly quoted or
    paraphrased. Furthermore, I certify that this assignment was prepared
    by me specifically for this course.
*********************************************************/
```

/!\ Note: The file name, student name and email ID must be changed per student.