CS2123 Program #2 Postfix Evaluation (40 points)

This is a continuation of Program #1 (Infix to Postfix). In this assignment, we evaluate the postfix expressions. I have provided a driver, include file, and data files.

This program is important in helping you understand

- Using a stack to evaluate postfix
- Integrating your code with someone else's code (mine)
- Techniques for error handling
- The use of unions.

Input:

There are two input files: course data and queries. See cs2123p2Driver.c for more information.

Process: (green highlighting indicates work you previously did and yellow highlighting indicates your new work):

- Read the course data (as in Program #0) and print the course data (as in program #0).
- Read the infix expressions as was done in Program #1. For each expression:
    a. Convert the infix to postfix (as in Program #1).
    b. Evaluate the postfix expression, producing an array of booleans (one entry for each course).
    c. Print the infix expression, the postfix form of the expression, and evaluated result. The driver does this.

Approaches for Evaluating Operators

Approach #1: Iterate over the list of courses and then evaluate the query for each course. The operator evaluation result is simply true or false (since only one course is considered).

- For N courses, this approach evaluates the same query N times.
- The operator result (i.e., a boolean) must be stacked.

Approach #2: When evaluating an operator, iterate over the courses to determine if each course satisfies the operator. The operator evaluation result is an array corresponding to the course array with a boolean indicating whether the course satisfies the result.

- A query is evaluated once.
- Each operator will iterate over the list of N courses.
- The operator result (i.e., the array of booleans) must be stacked.

We will use approach #1.

Additional functions provided by Larry:

```
void printQueryResult(Course courseM[], int iNumCourse, QueryResult resultM[])
       Prints the courses which have a corresponding element in the resultM array
       turned on.
int never(Course *pCourse, Attr attr)
       Determines whether a course has a particular attribute (type and value).
       If it doesn't, never returns TRUE; otherwise, it returns FALSE.
```

```
int getCourseData(Course courseM[])
       Gets course data and their corresponding attributes (type and values).
```

Use of union
Since the evaluation needs two types of elements, we will use **union** as shown in the include file.

Some Files for your use:

p2Query.txt - data file containing infix expressions which are evaluated as queries
p2Course.txt - course data file similar to program #0.
cs2123p2.h - include file for Program #2
cs2123p2Driver.c - driver program that I provided. It has several useful routines to help reduce your effort. Please review this code

Your coding:

- Create your code in **cs2123p2.c** (not cs2123p2Driver.c). Based on what the driver calls, you need to create (at least) these functions:

```
int convertToPostFix(char *pszInfix, Out out)
// you did this in program #1
       It returns 0 if it converted successfully.  Otherwise, it returns a
       value which indicates an error in
       the infix data (e.g., missing left paren, missing right paren)
       It populates the out array using the addOut function (provided in
       the driver).
void printCourseData(Course courseM[], int iNumCourse)
      This was done in program #0.
void evaluatePostfix(PostfixOut out, Course courseM[], int iNumCourse
       , QueryResult resultM[])
       This evaluates a postfix expression (which is in out) using the
       courseM array.  For each course satisfying the posftfix query,
       it sets its corresponding position in resultM to TRUE.
int atLeastOne(Course *pCourse, Attr attr)
       Determines whether a course has a particular attribute (type and
       value).  If it does, atLeastOne returns TRUE; otherwise, it returns
       FALSE.
int only(Course *pCourse, Attr attr)
       This looks at EACH attribute in the course's attrM array for the
       specified course.  If the attribute's szAttrType matches the
       specified szAttrType, then its corresponding szAttrValue must match
       the specfied attribute value.  If it doesn't match the attribute
       value, FALSE is returned.  only examines each of the attributes for
       the specified course.
You will also have to provide code for AND and OR.
```

- Modify the Makefile from program #1 to reference p2 for each of the files.

Requirements:

1. Your code must be written according to my programming standards.
2. You should not have to modify cs2123p2Driver.c. If you want to modify it, please see me before you do that.
3. Turn in your C code, include files, and output generated using the specified input file.
4. Make certain you free up allocated memory (e.g., stack).
5. Modularity matters.

Hint:

There are many useful functions in cs2123p2Driver.c. These can greatly simplify your code.

Sample Output (partial):

```
ID          Course Name
                      Attr       Value
CS1083      Intro I
                      LANG       JAVA
                      PROF       ROBBINS
                      PROF       HOLLAND
                      PROF       SHERETTE
                      PROF       ROBINSON
                      PROF       ZHU
                      RECIT      N
                      DEPT       CS
MAT2233     Discrete Math
                      PREREQ     MAT1214
                      PREREQ     CS1713
                      PROF       SHERETTE
                      PROF       TANG
                      RECIT      N
                      DEPT       MAT
MAT3333     Math Found
                      PREREQ     MAT1223
                      PREREQ     CS1713
                      PROF       SHERETTE
                      PROF       TANG
                      RECIT      N
                      DEPT       MAT
CS1713      Intro II
                      PREREQ     CS1083
                      LANG       C
                      PROF       CLARK
                      PROF       HOLLAND
                      PROF       SHERETTE
                      PROF       ROBINSON
                      PROF       ZHU
                      RECIT      Y
                      DEPT       CS
CS2123      Data Structures
                      PREREQ     CS1713
                      LANG       C
                      PROF       CLARK
                      PROF       SHERETTE
                      PROF       ROBINSON
                      PROF       KORKMAZ
                      PROF       TOSUN
                      RECIT      Y
                      DEPT       CS
CS3843      Comp Org
                      PREREQ     CS2123
                      LANG       C
                      LANG       IA32
                      PROF       CLARK
                      PROF       TIAN
                      PROF       MUZAHID
                      RECIT      Y
                      DEPT       CS
CS3723      Pgm Lang
                      PREREQ     CS3443
                      PREREQ     MAT2233
                      LANG       C
                      LANG       CPP
                      LANG       JAVA
                      LANG       LISP
                      LANG       PYTHON
                      PROF       CLARK
                      RECIT      N
                      DEPT       CS
CS3423     Sys Pgm
                      PREREQ     CS2123
                      PROF       MAYNARD
                      PROF       LAMA
                      PROF       CLARK
                      LANG       C
                      LANG       PERL
                      LANG       BASH
                      RECIT      Y
                      DEPT       CS
CS3443     App Pgm
                      PREREQ     CS2123
                      PROF       BYLANDER
                      PROF       ROBINSON
                      PROF       NIU
                      RECIT      N
                      DEPT       CS
                      LANG       JAVA
CS4713     Compiler
                      PREREQ     CS3723
                      PREREQ     CS3843
                      PROF       CLARK
                      RECIT      N
                      DEPT       CS
                      LANG       JAVA
CS3343     Anlys of Algo
                      PREREQ     CS2123
                      PREREQ     MAT3333
                      PREREQ     MAT2233
                      PROF       SHERETTE
                      PROF       GIBSON
                      PROF       BYLANDER
                      RECIT      Y
                      DEPT       CS
                      LANG       JAVA
Query # 1: RECIT = N
       RECIT N =
       Query Result:
       ID       Course Name
       CS1083   Intro I
       MAT2233  Discrete Math
       MAT3333  Math Found
       CS3723   Pgm Lang
       CS3443   App Pgm
       CS4713   Compiler
Query # 2: RECIT = Y
       RECIT Y =
       Query Result:
       ID       Course Name
       CS1713   Intro II
       CS2123   Data Structures
       CS3843   Comp Org
       CS3423   Sys Pgm
       CS3343   Anlys of Algo
Query # 3: PROF = CLARK
       PROF CLARK =
       Query Result:
       ID       Course Name
       CS1713   Intro II
       CS2123   Data Structures
       CS3843   Comp Org
       CS3723   Pgm Lang
       CS3423   Sys Pgm
       CS4713   Compiler
Query # 4: PROF NEVER CLARK
       PROF CLARK NEVER
       Query Result:
       ID       Course Name
       CS1083   Intro I
       MAT2233  Discrete Math
       MAT3333  Math Found
       CS3443   App Pgm
       CS3343   Anlys of Algo
Query # 5: PROF ONLY CLARK
       PROF CLARK ONLY
       Query Result:
       ID       Course Name
       CS3723   Pgm Lang
       CS4713   Compiler
Query # 6: PROF = CLARK AND RECIT = N
       PROF CLARK = RECIT N = AND
       Query Result:
       ID       Course Name
       CS3723   Pgm Lang
       CS4713   Compiler
```