Program 6 - Degree Plan

This program#5 and program #6 are part of a combined group programming assignment worth a total of 100 points.

We will use a double adjacency list graph to represent courses and their prerequisites:

* Each vertex represents course information
* Edges represent prerequisite relationships from a prereq to a successor
* Since it is a double adjacency list, each prerequisite relationship is two edge nodes. There is an edge node for each of these two lists:
  + For the successor, it is on a list of its prereqs
  + For the prereq, it is on a list of successors
* See cs2123p6.h

**Command** File

Input file stream contains multiple types of records (terminated

by EOF). Please use getToken to get the command and then use sscanf to read any associated data.

(New commands for Program 6):

PLAN szCourseId

This course is included in the set of courses we want to show in a semester by

semester plan. If the course doesn't exist, show a warning.

DOPLAN

This produces and prints the semester by semester plan. (See the sample output.)

DELETE szCourseId

Deletes the course, updates the edge lists appropriately, and frees the edge nodes. As a result of this command:

o This course will not be associated with any successors. (This also means that any successors will no longer have it as a prerequisite.)

o This course will not be associated with any prereqs. (This also means that no prereqs will have this course as a successor.)

PRTHASH

Prints the graph in a special format to show the hash table. This uses printHash which was provided by Larry. (See sample output.)

**Notes**

1. Copy your code into another folder, but do not copy cs2123p5.h. All your code needs to be compiled using cs2123p6.h so change your include file references.
2. Functions for program 6:

void **deleteCourse**(Graph graph, int iVertex)

This is invoked due to the DELETE command. This will delete a course:

* Because of the delete, this course will not be associated with any successors. (This also means that any successors will no longer have it as a prerequisite.)
* Because of the delete, this course will not be associated with any prereqs. (This also means that no prereqs will have this course as a successor.)
* Frees all edge nodes which reference this vertex.
* The course is marked deleted in the vertex array.
* As part of the extra credit, it is also removed from an overflow chain if it is in the overflow area.

void **setLevel**(Graph graph, Plan plan, int iVertex, int iLev)

Sets the vertex's semester level based on where that course should be taken (at an earliest). Prior to setting this value (and prior to calling setLevel), you probably want to determine the distance from the source for this vertex.

void **doPlan**(Graph graph, Plan plan)

Sets each semester in the plan. Although a course could have originally occurred in an earlier semester, only five courses are allowed per semester. If a course is pushed back, all of its successors have to be pushed back. There is a lot of flexibility in how this function can be coded.

The following functions are used in the extra credit code and are provided by Larry.

int **hash**(Graph graph, char szCourseId[])

Hashes a course ID to return a number between 0 and g->iOverflowBegin.

void **printHash**(Graph graph)

Prints the hash table in a format that helps show the primary area

and overflow area. For each entry in the primary area, it prints

the entry (if present) and prints corresponding synonyms.

1. More functions:
   * There are many more functions you need to create. Some of them will be called based on the commands. Others will be called by other functions.
2. **Hard-coding particular courses** (or **using other data to specify the sequence beyond what I provide**) will result in a **zero** on the entire assignment.

**What to turn in?**

Pgm 5 and pgm 6: via upload in BlackBoard

* team leader (for groups) must turn in the following as a single zip file (named with the group or individual name):
  + all .c files
  + .h file(s)
  + Makefile
  + output for each test file
  + instruction to TA sheet (explaining who is on the team (last name, first name) and how to compile/execute your code)
* Other participants in a group:
  + instruction sheet explaining who is on the team (last name, first name) and who is submitting the code
* "selfs" (teams of 1)must turn in the following as a single zip file(named for the individual as lastFirst.zip):
  + all .c files
  + .h file(s)
  + Makefile
  + output
  + instruction to TA sheet (explaining who is on the team (last name, first name) and how to compile/execute your code)

Hwk 6.1: via upload in Blackboard (everyone) due when pgm#6 is due

* Group Evaluation form as a PDF
* Peer Evaluation form as a PDF
* What I did form as a PDF

**Extra Credit** (8 points + 100/n)

1. Late submissions are not eligible for extra credit.
2. All functionality must be working to receive extra credit.
3. The vertexM array is still used; however, the entries with subscripts before OVERFLOW\_BEGIN are in the primary area. Entries beginning with OVERFLOW\_BEGIN are in the overflow area.
4. A hash function is provided and must be used so that we have consistent results.
5. Collisions are chained into an overflow area.
6. In your code prior to attempting hashing, you have many functions that referenced graph->iNumVertices. You may want to set that variable to MAX\_VERTICES to help your existing code.
7. There will be additional data files for the extra credit.

**Program #6 Sample Semester Plan Information**

>> \*

>> \* Program 6

>> \*

>> PLAN CS1083

>> PLAN CS1713

>> PLAN CS2123

>> PLAN MAT2233

>> PLAN CS3733

>> PLAN MAT3333

>> PLAN CS3343

>> PLAN CS3423

>> PLAN CS3443

>> PLAN CS3723

>> PLAN CS3843

>> PLAN CS3853

>> PLAN MAT1214

>> PLAN MAT1224

>> \*

>> \* Add the concentration - security CS3433 CS4353 CS4363

>> \*

>> PLAN CS3433

>> PLAN CS4353

>> PLAN CS4363

>> \* Get the plan

>> \*

>> DOPLAN

Semester Plan

Semester #1

CS1083 Intro I

MAT1214 Calculus I

Semester #2

CS1713 Intro II

MAT1224 Calculus II

Semester #3

MAT3333 Math Found

MAT2233 Discrete Math

CS2123 Data Structures

Semester #4

CS3343 Analysis of Algo

CS3843 Comp Org

CS3443 Appl Pgm

CS3423 Sys Pgm

Semester #5

CS3723 Pgm Lang

CS3853 Computer Arch

CS3733 Operating Systems

CS3433 Princ of Security

Semester #6

CS4363 Cryptography

CS4353 Unix & Net Security

**Program 6 Sample PRTHASH output**

>> COURSE CS3793 Artificial Intel

>> PREREQ CS3343

>> PRTHASH

Vx Crse Id Course Name Chain

Collision Vx Crse Id Course Name Chain

2 CS1713 Intro II -1

7 MAT3333 Math Found -1

8 CS3343 Analysis of Algo -1

9 CS3723 Pgm Lang -1

10 CS3733 Operating Systems -1

11 CS1083 Intro I 30

30 CS3743 Database Systems -1

12 CS3753 Intro Data Sci -1

14 CS3423 Sys Pgm 31

31 CS3773 Software Engineering -1

15 CS3433 Princ of Security -1

16 CS2123 Data Structures 32

32 CS3793 Artificial Intel 29

29 CS3443 App Pgm -1

19 CS3843 Comp Org -1

20 MAT1214 Calculus I -1

21 MAT1224 Calculus II -1

25 MAT2233 Discrete Math -1

Overflow free list

**33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59**