CS 6103D Software Systems Laboratory

PROBLEM 1B Evaluation **11.10.2021**

General Instructions

- The evaluation consists of two parts PART A and PART B. You will be allowed to proceed to PART B ONLY if you complete PART A and submit the code in EduServer.
- Design:
    - ■ Write the design ***only*** for **PART A** and submit it in the EduServer **before 2:30 PM**.
    - ■ No need to write a design for **PART B**.
- Implementation:
    - ■ Implement **PART A,** make sure that your program works correctly for the given sample I/O and submit code for PART A in the EduServer **before 3:15 PM**. **If you need more time for completing PART A, you may request your instructor for the same.**
    - ■ After submitting PART A, you may inform the instructor that you have submitted and then proceed with coding for PART B.
    - ■ No need to submit the source code in EduServer for **PART B**. You should complete the coding for PART B before **4.15 PM** and get the result verified by your evaluator before **4.45 PM**

Mark Distribution:

Maximum Marks - 7 marks

Design - 2 marks

Viva - 2 marks

Implementation - 3 marks (Part A - 1 + Part B - 2)

Modify the program developed for problem 1B as follows:

Part A

Modify the *regList*, such that each node has two fields - *name* (string) and *rollNo* (string). Maintain the list in the sorted order of *rollNo* by modifying the *insert()* function appropriately.

Design: Write algorithm (in pseudocode) for the required *insert()* function

Input/Output Format

The input should be read from a file '*input.txt*'.

The input file consists of multiple lines.

- The first line contains an integer *n>0*, the number of courses in a semester.
- The next *n* lines contain details of the *n* courses - in each line, *code, name,* and *credits* of a course, separated by a single space.
- The next set of lines indicate the operations to be performed. Each line begins with a character from *{i, d, p, s}* followed by zero or more string(s).
    - Character *i*: Character *i* followed by three strings - *rollNo, stud_name,* and *code* corresponding to the student roll number, name and course code respectively, separated by a space.
        - Call function *insert()* to insert a new node with the given *rollNo* and *stud_name* in its correct sorted position to the *regList* corresponding to the course *code.*
    - Character *d*: Character *d* followed by three strings - *rollNo, stud_name* and *code* corresponding to the student roll number, name and course code, separated by a space.
        - Call function *delete()* to delete the node with the given details from *regList* corresponding to the course *code.*
    - Character *p*: Character *p* followed by a string *code* corresponding to the course code.
        - Print the details of the course *code* as follows:
            - in the first line, print the course *code*, *name*, and *credits* (separated by a space).
            - in the next lines, print the details (*rollNo and stud_name)* of the students registered in the course, as per the order in *regList*.
            - if *regList* is empty, print "No students enrolled for this course"
    - Character s: Terminate the program

Sample Input (file *input.txt*)

```
4
CS6101D MFC 4
CS6111D ALG 4
CS6213D FIS 4
CS6103D SSL 1
i B210023CS SARITHA CS6103D
i B210014CS NEHA CS6103D
i B210001CS ALI CS6213D
p CS6101D
i B210014CS NEHA CS6213D
i B210021CS RIA CS6111D
d B210014CS NEHA CS6213D
p CS6213D
```

```
i B210001CS ALI CS6101D
i B210025CS SAMEER CS6101D
i B210023CS SARITHA CS6101D
i B210002CS ANCY CS6213D
i B210010CS JOHN CS6213D
i B210021CS RIA CS6213D
i B210023CS SARITHA CS6213D
d B210021CS RIA CS6213D
d B2100253CS SAMEER CS6101D
d B210001CS ALI CS6213D
p CS6101D
p CS6213D
p CS6111D
p CS6103D
s
```

Output

```
CS6101D MFC 4
 No students enrolled for this course
CS6213D FIS 4
B210001CS ALI
CS6101D MFC 4
B210001CS ALI
B210023CS SARITHA
CS6213D FIS 4
B210002CS ANCY
B210010CS JOHN
B210023CS SARITHA
CS6111D ALG 4
B210021CS RIA
CS6103D SSL 1
B210014CS NEHA
B210023CS SARITHA
```

Part B

Modify the *course struct* by adding new fields - *type* (string), *maxLimit* (int). The possible values of the two fields are:

- *type* - either “core” or “elective”
- *maxLimit* - 0 for “core”, >0 for “elective”.

Modify your program such that all students can register for core courses. The *waitList* is required only for elective courses.

Assume that the *insert()* is the same as the one specified in PART A.

Input/Output Format

The input should be read from a file ‘*input.txt*’.

The input file consists of multiple lines.

- The first line contains an integer *n>0*, the number of courses in a semester.
- The next *n* lines contain details of the *n* courses - in each line, *code, name, credits*, *type*, and *maxLimit* of a course, separated by a single space.
- The next set of lines indicate the operations to be performed. Each line begins with a character from *{i, d, p, s}* followed by zero or more string(s).

    - ▪ Character *i*: Character *i* followed by three strings - *rollNo, stud_name,* and *code* corresponding to the student roll number, name, and course code respectively, separated by a space.
        - ▪ If the *type* of the course is ‘elective’:
            - if the number of students in *regList* corresponding to course *code* is *maxLimit*, insert the given *rollNo* and *stud_name* into the queue *waitList* corresponding to course *code.*
            - Otherwise, call *insert()* and insert a new node with the given *rollNo* and *stud_name* into the *regList* corresponding to the course *code.*
        - ▪ If the *type* of the course code is ‘core’, call *insert()* and insert a new node with the given *rollNo* and *stud_name* into *regList* corresponding to the course *code.*
    - ▪ Character *d*: Character *d* followed by three strings - *rollNo, stud_name,* and *code* corresponding to the student roll number, name, and course code, separated by a space.
        - ▪ Call function *delete()* and delete the node with the given student details from the *regList* of the course *code.*

    - If *waitList* corresponding to the course *code* is not empty, dequeue a node from *waitList* in FIFO order and insert the node to *regList* corresponding to the course *code.*
- Character *p*: Character *p* followed by a string *code* corresponding to the course code.
    - Print the details of the course *code* as follows:
        - in the first line, print the course *code*, *name*, *credits, type,* and *maxLimit* (separated by a space).
        - in the next lines, print the details (*rollNo and stud_name)* of the students registered in the course, as per the order in *regList.*
        - If *type* of the course *code* is "elective"
            - in the next line, print "WAITLISTED STUDENTS"
            - in the next lines, print the details (*rollNo and stud_name)* of the students as per the order (from *head* to *rear*) in the queue *waitList*, if *waitList* is empty print "No students in the waiting list"
- Character s: Terminate the program

Sample Input (file *input.txt*)

```
4
CS6101D MFC 4 core 0
CS6111D ALG 4 core 0
CS6213D FIS 4 elective 3
CS6103D SSL 1 core 0
i B210014CS NEHA CS6103D
i B210001CS ALI CS6103D
i B210023CS SARITHA CS6103D
i B210021CS RIA CS6103D
p CS6103D
i B210001CS ALI CS6213D
i B210021CS RIA CS6213D
i B210014CS NEHA CS6213D
d B210021CS RIA CS6213D
p CS6213D
i B210023CS SARITHA CS6213D
i B210004CS JOHN CS6213D
i B210010CS KEERTHI CS6213D
p CS6213D
d B210001CS ALI CS6213D
d B210004CS JOHN CS6213D
```

```
i B210021CS RIA CS6213D
p CS6213D
s
```

Output

```
CS6103D SSL 1 core 0
B210001CS ALI
B210014CS NEHA
B210021CS RIA
B210023CS SARITHA
CS6213D FIS 4 elective 3
B210001CS ALI
B210014CS NEHA
WAITLISTED STUDENTS
No students in the waiting list
CS6213D FIS 4 elective 3
B210001CS ALI
B210014CS NEHA
B210023CS SARITHA
WAITLISTED STUDENTS
B210004CS JOHN
B210010CS KEERTHI
CS6213D FIS 4 elective 3
B210010CS KEERTHI
B210014CS NEHA
B210023CS SARITHA
WAITLISTED STUDENTS
B210021CS RIA
```