CS F363 Compiler Construction
Assignment-1 (Question-2)

Due date: 07 March 2024 11:59 PM **Marks : 10** (5%)

You are given students' performance data in a text file (say *data.txt*), and the data is a collection of records. Each record starts with a $ followed by the student ID (studID) followed by a sequence of one or more course grade information. Note that studID is an alphanumeric string containing only upper case letters and digits 0 to 9; it starts with upper case alphabets followed by at least one digit from $0, 1, \ldots, 9$. Ex. SAM456, JAM123, CAT137, etc. Each course grade information is given in a 3-tuple < courseID units grade > where courseID is an alphanumeric string containing only upper case letters and digits 0 to 9; starts with upper case alphabets followed by at least one digit from $0, 1, \ldots, 9$. Ex. ABC12, X123, PQR13, etc., units is an integer in range 2 to 4, and grade is a letter grade from $\{A, A-, B, B-, C\}$. Note that at least one space exists between courseID and units and between units and grade.

The general structure of a (valid) record is given below

$studID < courseID units grade > < courseID units grade > $\cdots$ < courseID units grade >

(the tuples may be separated by spaces).

Please note the following:

1. A record starts with $ and is completely contained in the same line (do not spread over multiple lines). Hence, the information before the $ symbol in a line, if any, is not a part of any record. You can ignore such information.

   For example: <CSF363 4 A-> $ CAT123 <CSF211 4 A-><CSF111 3 B->

   In the above example, "<CSF363 4 A->" is not part of any record.

2. If studID is missing after $ or does not match the pattern as mentioned above (call the record as invalid record), then just ignore the record.

   For example, $ <CSF363 4 A><CSF111 3 B-> is an invalid record since studID is missing; hence, you just ignore the record.

   $ Abc123 <CSF363 4 A><CSF111 3 B-> is also an invalid record because "Abc123" is not matching the pattern of studID.

3. *data.txt* may contain some comments that start with //. In this case, just ignore the line starting from the appearance of //. Note that before //, there may be some valid records.

4. Further, more than one record can be placed in a single line. In this case, consider all such valid records.

   For example:

   $ CAT123 <CSF211 4 A-> <CSF111 3 B-> $ MAT897 <MATHF111 4 B>

5. If some information is missing in 3-tuple < courseID units grade >, then ignore the tuple, not the entire record.

   Example: In $ CAT123 <CSF211 4 A-><CSF111 B-><MATHF111 4 B>, just ignore the tuple <CSF111 B-> since units are missing.

   Further, if units is not in the range 2 to 4 or grade $\notin \{A, A-, B, B-, C\}$, then ignore the tuple.

Your task is to write a LEX program that finds the following :

1. The number of valid records
2. The number of comments
3. Given a student ID, find the number of distinct courses the student registered for and the CGPA of the student. I believe that you know the formula to compute CGPA.

To compute CGPA, use the following table:

grade	points
A	10
A-	8
B	6
B-	4
C	2

Table 1: Grades and points information

Further, for a student, if a course performance information is repeated more than once then consider the latest grade for the CGPA calculation.

Input format: Two text files will be given as part of the input:

1. data.txt: The details about the file are given above.
2. input.txt contains a single line with student ID (studID).

Output format: Generate a file, output.txt (do not use other names) with the output of your program in the format $@k_1@k_2@k_3@k_4\#$ where

1. k_1 is the number of valid records (non-negative integer) in the data.txt file.
2. k_2 is the number of comments (non-negative integer) in the data.txt file.
3. k_3 is the number of distinct courses (non-negative integer) the student registered for, whose ID is given in input.txt is given.
4. $k_4 = \text{round}(CGPA * 100)$; $CGPA$ is of the student whose ID is given in input.txt. Use round function provided by C language.

Submission guidelines:

1. Submit a single LEX file and name it with your group code. If your group code is BITS1234, your file name must be BITS1234.l, and it must be executable on the Ubuntu 23.04 system.
2. One submission per group is sufficient.
3. Strictly follow the input and output formats.
4. **Late submission policy:** Each 1hr delay will fetch 1% penalty of the total marks obtained, and late submissions will not be accepted after 48 hours from the due date.