CSL302: Compiler Design

Lab Exam-1 (2025-26-M Semester)

Max. Points: 100 **Duration:** 1 hour 30 minutes

September 11, 2025

Instructions

- The question paper contains 2 questions. Prepare 2 folders; each one should corresponding to question and name the folder as <Q1> or <Q2> depending on the question number.
- The solution to each question should contain a ReadMe file, which should list the instructions to execute your program.
- Prepare a zip file for your solutions.
- The submission details will be announced at the end of the exam.

Question-1

Assume that the IIT Bhilai ID card has a unique ID number consisting of 8 digits as the following.

D1	D2	D3	D4	D5	D6	D7	D8

Details of the Digits:

- The digit D1 represents the type of cardholder as per the following.

Value of D1	**Type of the Card Holder**
1	Regular Student
2	Visiting Student
3	Regular Employee
4	Contract Employee
5-9	Other categories
0	Invalid

- D2 and D3 denote the year of admission for students and year joining for employees. Each of D2 and D3 take values from 0 to 9.
- D4 through D7 denote the serial number within the group as per the following.

1. For students (i.e., D1 is either 1 or 2)

Serial Number	**Programme**
0000-1999	Ph.D students
2000-3999	Master students
4000-9999	Under graduate students

2. For employees, it denotes the serial number from 0000 to 9999.

- D8 denotes the dependent family information. For employees, the following table is to be used for D8. For students, D8 should be 0; any other value is considered invalid.

Value	**Semantics**
0	Self
1-9	Dependent

Given the above information, write a lexical analyzer that, given a character stream, it prints the tokens corresponding to it as per the following.

Type of Person	**TOKEN**
Ph.D students	PHD_STUDENT
Master students	MASTER_STUDENT
Under graduate students	UG_STUDENT
Regular employee	EMPL_REG
Dependent of regular employee	EMPL_REG_DEP
Contract employee	EMPL_CON
Dependent of contract employee	EMPL_CON_DEP
Other category type	OTHER
Any invalid ID. For example, invalid characters, incorrect length	INVALID

[40 Points]

Question-2

Design and implement a Lexical Analyzer (scanner) that reads assembly language code and converts it into a sequence of tokens. The features are listed below.

1. **Opcodes (instructions)** - Recognize a fixed set of mnemonics: `MOV, ADD, SUB, MUL, DIV, LOAD, STORE, JMP, CMP, HALT`.
2. **Labels** - User-defined names containing one or more captial alphabets (e.g., `LOOP`).
3. **Registers** - Recognize register names: `R0, R1, R2, ..., R9`.
4. **Immediate Numbers** - Integer values beginning with # (e.g., `#10`, `#255`, `#-5`).
5. **Delimiters / Special Symbols** - `,`, :
6. **Comments** - Everything after `;` on a line should be ignored (including ;).
7. **Errors** - Anything other than above tokens should be treated as error.

Example Input

```
START: MOV R1, #10      ; load 10 into R1
ADD R2, R1        ; R2 = R2 + R1
SUB R3, #5
JMP START
HALT
```

Expected Output

```
LABEL       : START
DELIMITER   : :
OPCODE      : MOV
REGISTER    : R1
DELIMITER   : ,
IMMEDIATE   : #10
OPCODE      : ADD
REGISTER    : R2
DELIMITER   : ,
REGISTER    : R1
OPCODE      : SUB
REGISTER    : R3
DELIMITER   : ,
IMMEDIATE   : #5
OPCODE      : JMP
LABEL       : START
OPCODE      : HALT
```

[60 Points]