DR B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR, PUNJAB (INDIA)

Department of Computer Science and Engineering

Assignment-1 of SPCD-401

DATE OF ASSIGNMENT: - 19th Sept, 2024

DATE OF SUBMISSION: - 30th Sept, 2024

Please take note of the following instructions for submitting your assignments:

Physical Mode:

- Assignments need to be submitted both in physical mode and also uploading it to google classroom drive.
- Ensure your assignment is neatly written, legible, and well-organized.
- Label your assignment with your name, course name and roll no.
- Staple or bind multiple pages together.
- Submit your assignment on or before the due date. Late submissions may be subject to penalties.

Questions (1) :- Consider the following assembly program:

```
START   1000
LOOP    LDA    NUM1
        ADD    NUM2
        STA    RESULT
        JMP    LOOP
NUM1    DC     25
NUM2    DC     30
RESULT DS      1
END
```

1. **First Pass:**

- Construct the symbol table after the first pass of the assembler.

- Describe the role of the first pass in a two-pass assembler process.

2. **Second Pass**:

- Provide the machine code generated after the second pass of the assembler.
- Explain the purpose of the second pass and how it differs from the first pass in a two-pass assembler.

Question (2): - A student wrote two context-free grammars G1 and G2 for generating a single C-like array declaration. For example,

float f [10] [3];

The grammars use D as the start symbol, and use six terminal symbols: -
float ; id [ ] num.

Grammar G1	**Grammar G2**
D → int L;	D → int L;
L → id [E	L → id E
E → num]	E → E [num]
E → num] [E	E → [num]

Create parse tree using each grammar to generate the declaration mentioned above.

Question (3): -

a) Consider the following grammar and eliminate left recursion-

A → Ba / Aa / c

B → Bb / Ab / d

b) Consider the following grammar and eliminate left-factoring-

S → a / ab / abc / abcd

Question (4): - Consider the grammar with non-terminals N={S,C,S1}, terminals T={a, b, i, t, e}, with S as the start symbol, and the following set of rules:

$$S \rightarrow iCtSS_1 \mid a$$

$$S_1 \rightarrow eS \mid \epsilon$$

$$C \rightarrow b$$

Construct the LL (1) parsing table and check whether given grammar is LL (1) or not?

Question (5): Consider the grammar with non-terminals N={E,T,F}, terminals T={ (,) , +,*, id}, with E as the start symbol, and the following set of rules:

E→E+T/T

T→T*F/F

F→(E)/id

Construct the SLR(1) parsing table and check whether given grammar is SLR(1) or not? Also implement the stack for the **input string: id*id+id**

Question (6): Consider the grammar with non-terminals N = {S, A}, terminals T= {a, b}, with S as the start symbol, and the following set of rules:

S → CC

C → cC/d

Construct the LALR (1) parsing table and check whether given grammar is LALR (1) or not?
