1. Write a program to implement an operator precedence parser. The parser should be able to parse string |- id - id \* id + id-|. (Generate precedence table for the given operator only).
2. Given the following grammar write a C/C++ program to implement recursive decent parser that parses a string x + x \* x

E::= T {+T}\*

T::= V {\*V}\*

V::= <id>

1. Given the following grammar and parse table write a program to implement LL(1) parser that parses a string |- a + a \* a -|

X::= YX′

X′ ::= +Y X′ | ∑

Y::= Z Y′

Y′::= \*Z Y′ | ∑

Z::= a

1. Generate a DFA to validate a regular expression **a+(c/d)\*bbd\***. Write a program that uses this DFA and validates whether an entered string is valid or not.
2. Write a program to implement an operator precedence parser. The parser should be able to parse string |- id - id \* id + id-|. (Generate precedence table for the given operator only).
3. Given the following grammar write a C/C++ program to implement recursive decent parser that parses a string a – a /a

E::= T {-T}\*

T::= V {/V}\*

V::= b

1. Given the following grammar and parse table write a program to implement LL(1) parser that parses a string |- i - i / i -|

X::= YX′

X′ ::= -Y X′ | ∑

Y::= Z Y′

Y′::= /Z Y′ | ∑

Z::= a

1. Generate a DFA to validate a regular expression **aa+(c/d)\*b(bd)\***. Write a program that uses this DFA and validates whether an entered string is valid or not.
2. Generate a DFA to validate a regular expression **aba+b(bd)\***. Write a program that uses this DFA and validates whether an entered string is valid or not.