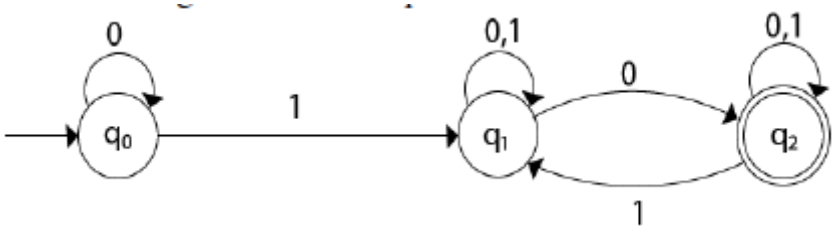
Tutorial Questions
ATCD

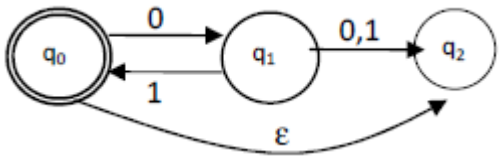
UNIT-1

1. Why do we need to study automata theory and formal languages?
2. What is DFA? Find the DFA for the following languages over {0, 1}*
   i) The set of all strings such that number of 0's is odd
   ii) The set of all strings that contain exactly three 1's
   iii) The set of all strings that do not contain 1101
3. Design DFA which accepts language L={0,000,00000…….} over {0}
4. Design a DFA L(M) = {w | w ε {0, 1}*} and W is a string that does not contain three consecutive 1's.Process the string **0011001** on machine.
5. Design DFA to accept strings with 'c' and'd' such that number of d's are divisible by 4.Show the moves of DFA for the string **cdcddcd**
6. Design a DFA to accept the language
   L = {w |w has both an even number of 0's and an even number of 1's}.
   Represent obtained DFA by transition table. Show the transitions of DFA for the string **110101.**
7. What is DFA? Find the DFA's accepting for the following languages over the alphabet {a,b,c}:
   i) The set of all strings with **abc** as a substring
   ii) The set of all strings ending in **bac**
   iii) The set of all strings starting with **cab**
8. Draw a deterministic and non-deterministic finite automata for ∑= {A-Z} which accept a string containing "CSE" at the end of a strings of {A-Z}.
9. Design DFA for the following over {a,b}.
   i) All string containing not more than three a's.
   ii) All strings that has at least two occurrences of b between any two occurrences of a.
   iii) All strings ending with aa
10. What is DFA? Construct a DFA accepting the language
    { W ∈ {a,b}*| W has neither aa nor bb as substring}
11. Draw the DFA for the following
    i) To accept decimal strings divisible by 3 over the alphabet ∑ = {0,1,2,3,4,5,6,7,8,9}
    ii) To accept odd number of a's and even number of b's over alphabet ∑ = {a,b}
12. Design a DFA that reads strings made up of letters in the word 'CHARIOT' and recognizes these strings that contain the word 'CAT' as a substring.
13. Define Deterministic and Non-deterministic finite automaton.
14. Differentiate between NFA and DFA?
15. Design an NFA with Σ = {0, 1} accepts all string in which the third symbol from the right end is always 0.
16. Construct an NFA that accepts the set of all strings over {0,1} that start with 0 or 1 and end with 10 or 01.
17. Design a NFA for the following language L={0101^n where n>0}
18. Describe the procedure of converting NFA to DFA with a suitable example.

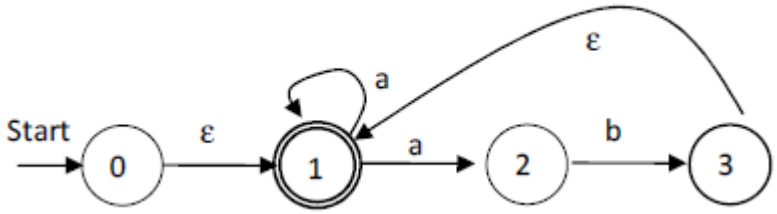
19. Convert the given NFA to equivalent DFA



20. Construct a DFA equivalent to the NFA given below



21. Define NFA with ∈ - moves and give example.
22. Depict the steps in converting an NFA with ∈ into NFA without ∈ with an example.
23. Show with an example equivalence between NFA with and without Є-transitions
24. Convert the following NFA-ε to NFA



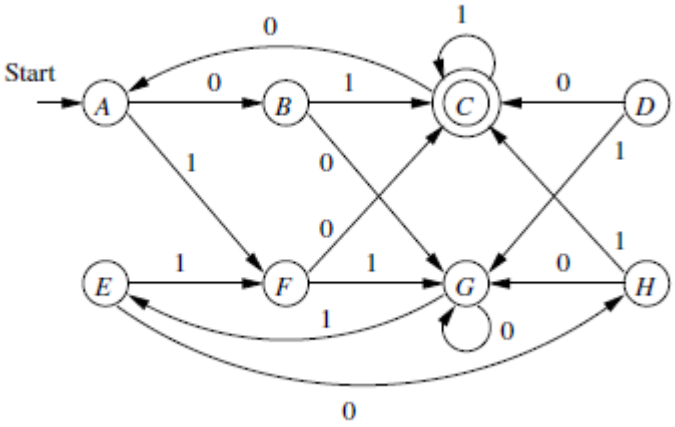
25. Explain the procedure for constructing minimum state DFA with an example.
    (OR) What is minimal DFA? Write the minimization Algorithm for DFA?
26. Let ∑= {a, b}, Give DFA that accepts any string with *aababb* as a substring. Minimize the DFA obtained.
27. Reduce the following DFA where q1 is the start state and q6 is the final state.

δ	0	1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_6	q_7
q_4	q_4	q_5
q_5	q_6	q_7
q_6	q_4	q_5
q_7	q_6	q_7

28. Construct Minimum state Automata for the following DFA?

δ	0	1
→ q1	q2	q6
q2	q1	q3
*q 3	q2	q4
q 4	q4	q2
q5	q4	q5
*q6	q5	q4

29. Reduce the DFA given below



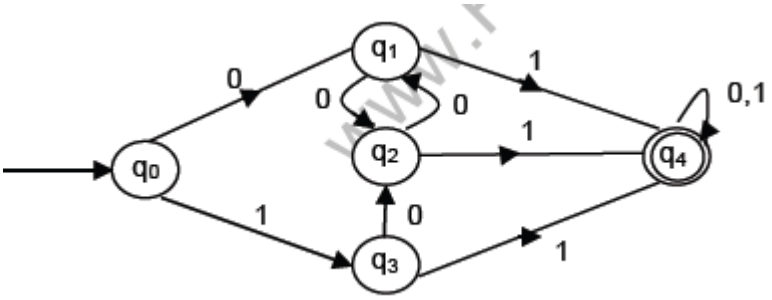
30. Construct Minimum state Automata for the following DFA

	0	1
→ A	B	A
B	A	C
C	D	B
$*D$	D	A
E	D	F
F	G	E
G	F	G
H	G	D

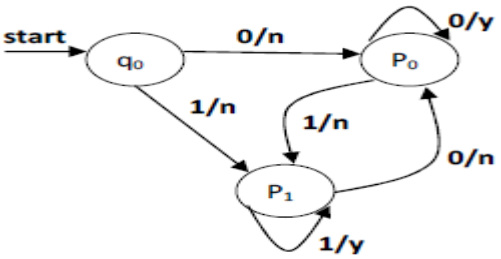
31. Construct Minimum state Automata for the following DFA

	0	1
→ A	B	E
B	C	F
$*C$	D	H
D	E	H
E	F	I
$*F$	G	B
G	H	B
H	I	C
$*I$	A	E

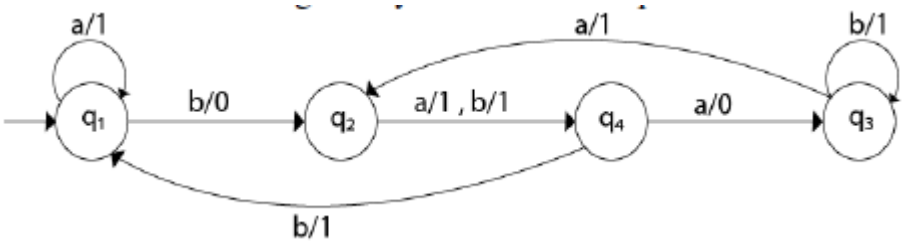
32. Minimize the finite automaton shown in figure below.



33. Bring out the differences between Moore and Mealy machines?
34. Design a Moore machine with the input alphabet {0, 1} and output alphabet {Y, N} which produces Y as output if input sequence contains 1010 as a substring otherwise, it produces N as output.
35. Design a mealy machine to print out 1's complement of an input bit string?
36. Design a Moore machine for 2's complement of binary number.
37. Construct the Moore machine to compute residue modulo 5 and finds to its equivalent Mealy machine.
38. Draw a Moore machine for calculating mod 3 of a given binary number. Find its equivalent mealy machine.
39. Construct a Moore machine that determines whether an input string contains an even or odd number of 1's. The machine should give 1 as output if an even number of 1's is in the string and 0 otherwise.
40. Design a Moore and Mealy machines for a binary input sequence such that if it has a substring 101, the machine outputs A, if the input has substring 110, it outputs B otherwise it outputs C.
41. Design a Moore and Mealy machine that scans sequence of input of 0 and 1 and generates output 'A' if the input string terminates in 00, output 'B' if the string terminates in 11, and output 'C' otherwise.
42. Design Moore and Mealy Machine to increment binary number by 1
43. Give Mealy and Moore machine for the following: For input from ∑*, where ∑= {0,1,2}, print the residue modulo 5 of the input treated as a ternary (base 3, with digits 0, 1 and 2) number.
44. Design a Moore Machine that will read sequences made up of letters a, e, i, o, u and will give as output, same characters except when an 'i' is followed by 'e', it will be changed to 'u'.
45. Design a Mealy machine to add two binary numbers of the form x1x2…xk, y1y2…yk?
46. Convert the following Mealy machine to an equivalent Moore machine

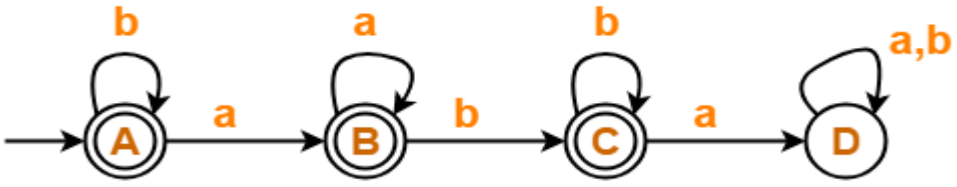


47. Convert the following Mealy machine into equivalent Moore machine.

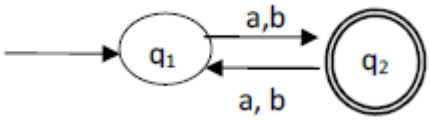


UNIT-2

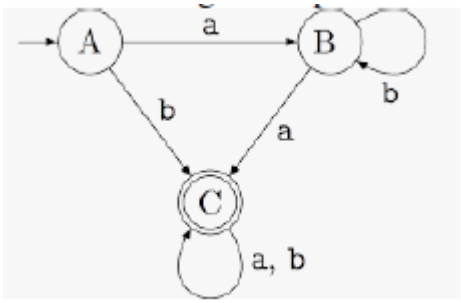
1. Define regular expression. Give regular expression for the following languages.
   (i) Strings over the alphabet {a, b} ending with ab
   (ii) Strings over the alphabet {0,1} that contain substring 10
   (iii) Strings over the alphabet {0,1} that contain 1 in the 3rd position from right end
2. What is regular expression? Write the regular expression for the following languages over {0, 1}*
   i) The set of all strings such that number of 0's is odd
   ii) The set of all strings that contain exactly three 1's
   iii) The set of all strings that do not contain 1101
3. Write regular expressions for the following language over the alphabet ∑= {0, 1}
   i) Strings with three consecutive 1's
   ii) Strings with three 1's
4. Write the regular expression for the language L over Σ = {0, 1} such that all the strings
   i) do not contain the substring 01.
   ii) should have at least one 0 and at least one 1.
5. Explain pumping lemma for regular languages with the applications of pumping lemma.
6. Prove that the language L = $\{(10)^p 1^q \mid p, q \in N, p \geq q\}$ is not regular.
7. Describe the closure properties of Regular sets.
   (or) Summarize the closure properties of regular language.
8. Explain about the identity rules of Regular Expressions?
   (or) List any ten algebraic laws for regular expressions and explain
   (or) Explain about the Properties of Regular Expressions?
9. Construct a DFA for the Regular expression (0+1)* (00+11) (0+1)* ?
10. Design a NFA for the given regular expression 1 (1* 01* 01*)*.
11. Write the steps to construct regular expression from given DFA?
12. Construct a NFA equivalent to the regular expression 10(0+11)0*1?
13. Construct an NFA for r = $(a+bb)^*ba^*$
14. What is relationship between finite automata and regular expression? Explain the process of converting DFA to regular expression.
15. Construct an NFA for the regular expression (a+b)* (aa+bb) (a+b)*
16. Convert the regular expression (((00)*(11)) + 01)* into an NFA.
17. Convert the following DFA to a regular expression.



18. Construct a regular expression for the given transition diagram



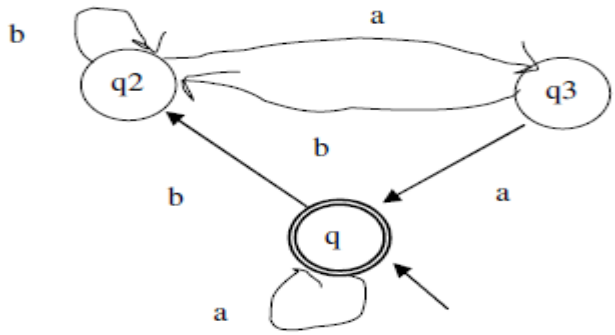
19. Construct a Regular expression corresponding to the following finite automata.



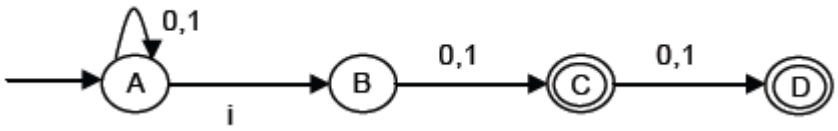
20. Construct a regular expression corresponding to the DFA represented by the below transition table. q_1 is both the initial state and final state.

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_2

21. Construct a NFA equivalent to the regular expression **(10+11)*00.**
22. Explain the procedure for converting finite automata to regular grammar with an example.
23. State and Explain the pumping lemma for Regular languages.
24. Prove that the following language L is not regular using pumping lemma
    L = { a^{2n} b^{3n} a^n | n≥0}
25. Write the regular expression for the L={w Є_{0,1}* | w has no pair of consecutive zeros?
26. (0/1)*011 for this regular expression draw the NFA with ϵ-transitions and convert it into NFA.
27. Give a regular expression that generates the language L over the alphabet Σ={a, b} where each b in the string is followed by exactly one or three a's.
28. Show that L={a^{2n} /n>0} is Regular.
29. What is a regular language? Convert the given regular expression to regular language.
    i) (1+ε)(00*1)0* ii) (0*1*)000(0+1)* iii) (00+10)*1*(10+00)*
30. Construct a Regular expression corresponding to the following finite automata.



31. Convert the following NFA into regular expression.



32. Explain about the Closure Properties of Regular sets?
33. Prove that the following language L is not regular using pumping lemma
    L = { *w* belongs to {a,b}* | $w = w^R$}
34. Design CFG for the following languages?
    i) L = { $a^m b^n$ /n, m>0,m = n}
    ii) L = { $a^m b^n$ /n, m>0,m>n}
    iii) L = { $a^m b^n$ /n, m>0,m<n}
    iv) L = { $a^m b^n$ /n, m>0,m ≠ n}
35. Design CFG for the following languages?
    i) L = { $N_a(w) = N_b(w)$ / w $\{a,b\}^+$}
    ii) L = { $N_a(w) > N_b(w)$ / w $\{a,b\}^+$}
    iii) L = { $N_a(w) < N_b(w)$ / w $\{a,b\}^+$}
    iv) L = { $N_a(w) \neq N_b(w)$ / w $\{a,b\}^+$}
36. Design CFG for the following languages?
    i) L = { $a^n b^n c^m$ /n, m>=1}
    ii) L = { $a^n b^m c^n$ /n, m>=1}
    iii) L = { $a^m b^n c^n$ /n, m>=1}
37. Design CFG for the following languages?
    i) L = { $a^n b^{n+2}$ /n>=0}
    ii) L = { $a^n b^{n-2}$ /n >=0}
    iii) L = { $a^{2n+3} b^n$ /n>=0}
    iv) L = { $a^n b^{2n}$ /n>=0}
    v) L = { $a^{2n} b^n$ /n>=0}
38. Design CFG for the following languages?
    i) L = { a^n /n>=1}
    ii) L = { $(ab)^n$ /n >=1}
    iii) L = { $(abc)^n$ /n>=1}
39. Design CFG for the following languages?
    i) L = { $a^n b^m c^m d^n$ /n, m>=1}
    ii) L = { $a^n b^n c^m d^m$ /n, m>=1}
40. Prove that **S →aSbS | bSaS |ε** is ambiguous.
41. For the Grammar {S→AS/a, A→SbA/SS/ba} construct Left most derivation and rightmost derivation for the string **aabbaaa?**
42. Define Ambiguous Grammar? Check whether the grammar S→aAB, A→bC/cd, C→cd, B→c/d Is Ambiguous or not?
43. Obtain GNF for the grammar S→AB, A→BS/b , B→SA/a ?
44. Obtain the Chomsky normal form for the following grammar E→E+T/T, T→a/CE?

45. Construct a derivation tree for the string **abcd** from the grammar
    S→aAB, A→bC, B→d, C→cd
46. Show that L= {a^p /p is prime} is Context free?
47. Construct CNF for the Grammar S→ABC, A→0B,B→CD/0,C→1
48. Write the general procedure to transform a grammar to Greibach Normal Form?
49. Remove Null production from the following grammar
    S →ASA | aB | b
    A → B
    B →b | ∈
50. Define Context Free Grammar. State and explain the closure properties of CFG.
51. Consider the CFG with {S,A,B} as the non-terminal alphabet, {a,b, ε} as the terminal alphabet, S as the start symbol and the following set of production rules
    S→ASA | aB
    A→B|S
    B→b|ε
    Convert the given grammar into CNF
52. Consider the CFG with {S,A,B} as the non-terminal alphabet, {0,1} as the terminal alphabet, S as the start symbol and the following set of production rules
    S →A1B
    A →0A / ∈
    B → 0B / 1B / ∈
    For the string w = 00101, find the Leftmost derivation, Rightmost derivation, and Parse Tree.
53. Show that language L={a^n $b^n c^n$ |n≥0} is not a Context Free.

54. Consider the CFG with {S,A,B} as the non-terminal alphabet, {a,b} as the terminal alphabet, S as the start symbol and the following set of production rules
    S →aB|bA
    S → aS|bAA| a
    B →bS| aBB| b
    Is this grammar ambiguous or unambiguous? Give justification to your answer.
55. Consider the CFG with {S,X,Y} as the non-terminal alphabet, {m,n,o} as the terminal alphabet, S as the start symbol and the following set of production rules
    S → XY | Xn | p
    X →mX | m
    Y →Xn | o
    Convert the given CFG into Greibach Normal Form
56. Generate left most and right most derivation and parse tree for given grammars
    G1: S→0B|1A, A→0|0S|1AA, B→1|1S|0BB for the string ***00110101***
    G2: S→Ab|bA, A→a|aS|bAA, B→b|bS|aBB for the string ***aaabbabbba***

57. Find equivalent grammar in CNF for S→bA|aB, A→bAA|aS|a, B→aBB|bS|b

58. Design CFG for the following language
    L = {0^n1^n | n ≥1}
    Give leftmost and rightmost derivations for a string 000111 in obtained grammar.

UNIT-3

1. Construct a PDA for L={a^nb^n /n ≥1}.Draw transition diagram.Using the instantaneous description notation process the string **aaabbb**
2. Define PDA and instantaneous description of PDA. Obtain a PDA to accept the language L={ wcw^R : w ∈{a,b}*}.Draw transition diagram of PDA. Show the moves by this PDA for string **abbcbba**
3. Explain the various ways of determining the acceptability of Pushdown Automata.
4. Construct a PDA that accepts L = {0^n1^n | n ≥0}
5. Design a PDA to accept the language of balanced parenthesis.
6. Design PDA for the following languages by Empty Stack?
   i) L = { $a^n b^n c^m$ /n, m>=1}
   ii) L = { $a^n b^m c^n$ /n, m>=1}
   iii) L = { $a^m b^n c^n$ /n, m>=1}
7. Design PDA for the following languages by Empty Stack?
   i) L = { $a^m b^n c^{m+n}$ /n, m>=1}
   ii) L = { $a^m b^{m+n} c^n$ /n, m>=1}
   iii) L = { $a^{m+n} b^m c^n$ /n, m>=1}
8. Design PDA for the following languages by Empty Stack?
   i) L = { $a^n b^m c^m d^n$ /n, m>=1}
   ii) L = { $a^n b^n c^m d^m$ /n, m>=1}
9. Design PDA for the following languages by final state?
   i) L = { $a^n bc^n$ /n>=1}
   ii) L = { $ab^n c^n$ /n>=1}
   iii) L = { $a^n b^n c$ /n>=1}
10. Design PDA for the following languages by final state?
    i) L = { $a^n b^3 c^n$ /n>=1}
    ii) L = { $a^3 b^n c^n$ /n>=1}
    iii) L = { $a^n b^n c^3$ /n>=1}
11. Design PDA for the following languages by final state?
    i) L = { $a^n bc^{2n}$ /n>=1}
    ii) L = { $ab^n c^{2n}$ /n>=1}
    iii) L = { $a^n b^{2n} c$ /n>=1}
12. Design PDA for the following languages by final state?
    i) L = { $a^n b^{n+1}$ /n>=1}
    ii) L = { $a^n b^{2n+1}$ /n>=1}
    iii) L = { $a^{2n} b^{3n}$/n>=1}
    iv) L = { $a^{3n} b^{2n}$/n>=1}
13. Design PDA for the following languages by final state?
    i) L = { $a^m b^n$ /n, m>0,m ≠ n}
    ii) L = { $a^m b^n$ /n, m>0,m>n}
    iii) L = { $a^m b^n$ /n, m>0,m<n}
14. Design PDA for the language

$$L(G) = \{a^n b^m a^k b^n | n, m, k \in \mathbb{N}\}.$$

15. Construct a PDA, M equivalent to the following CFG S→0BB, B→0S/1S/0, test whether 010^4 is in N(M)?
16. Construct a PDA for L={ wcw^R /w ∈(0+1)*}
17. Construct a PDA for the following grammar: S→AA/a, A→SA/b
18. Convert the grammar S→0AA, A→0S/1S/0 to a PDA that Accepts the same Language by Empty Stack?
19. Design a non deterministic push down automata for the following languages
    L1={a^nb^n |n>=0}, L2={ww^R |w in (0+1)*}
20. Convert the following grammar to a PDA that accepts the same language.
    G = (V, T, R, S) with V = {S}, T = {a, b, c}, and R = {S → aSa, S → bSb, S →c}.
21. Describe the components of Push Down Automata.
22. Construct a PDA from the following CFG.
    G = ({S, X}, {a, b}, P, S) where the productions are given below.
    S → XS | ∈ ,
    A → aXb | Ab | ab
23. Design a PDA for accepting a language {a^nb^{2n} | n>=1}.Show the moves of the PDA for the string **aabbbb**
24. Design a PDA for accepting a language {$a^{2n}b^n$ | n>=1}.Show the moves of the PDA for the string **aaaabb**
25. Differentiate between Deterministic PDA and Non-deterministic PDA.
    (or) Distinguish between a DPDA and NPDA
26. Define Push Down Automata. Explain the basic structure of PDA with a neat graphical representation.
27. Construct a PDA for recognizing the language L = {$a^ib^jc^k$ /i,j,k €N,i+k=j}.
28. Construct a PDA for recognizing the language of all strings over the input alphabet {a,b} such that the number of b's in each string are equal the number of a's. Show the moves of the PDA for the string **ababbbaa**
29. Construct a PDA for recognizing the language of all strings over the input alphabet {a,b} such that the number of b's in each string are twice the number of a's. Show the moves of the PDA for the string **abbabbbba**
30. Design PDA for recognizing the language of palindromes over the alphabet {0,1}.Draw the computations tree showing all possible moves for the strings **00100** and **00101**
31. When do you say that a language is recognized or accepted by a PDA? Design a PDA for **L={$a^ib^jc^k$/j>=i+k and i,j,k>0.}** Process the string **aaabbbbbbcc** using instantaneous description.
32. Design PDA for recognizing the language L={a^ib^j / j<=i and i,j >0} Show the moves of the PDA for the string **aaabb**
33. Does push down automata have memory? Justify your answer. Mention the applications of PDA.
34. Construct PDA for the given CFG, and test whether 010^4 is acceptable by this PDA.
    S → 0BB
    B → 0S | 1S | 0

35. Construct a Turing Machine for language L = $\{0^n1^n/n\geq1\}$
36. Construct a Turing Machine for language L = $\{0^n1^n2^n/n\geq1\}$
    (or) Construct a Turing Machine to recognize the Language $\{a^nb^nc^n/n\geq1\}$
    (or) Design a Turing Machine to recognize the language $\{1^n2^n3^n/n\geq1\}$
37. Design a Turing Machine to recognize the language $\{0^n1^n0^n / n\geq1\}$
38. Construct a Turing Machine for language L = $\{ww^R \mid w \in \{0, 1\}^*\}$.
    (or) Design a Turing Machine to recognize even palindromes over {0,1}
39. Design a Turing Machine to recognize odd palindromes over {0,1}
    (or) Construct a Turing Machine for language L = $\{w(0+1)w^R \mid w \in \{0, 1\}^*\}$.
40. Design a Turing Machine for recognizing L = $\{wcw \mid w \in \{a, b\}^*\}$.
41. Design a Turing Machine for recognizing L = $\{wcw^R \mid w \in \{a, b\}^*\}$.
    Show the moves of the TM for the input string **abbcbba.**
42. Design a Turing Machine for recognizing L = $\{xx \mid x \in \{a, b\}^*\}$. Show the moves of the TM for the input strings **abaaba** and **abaabb.**
    (or) Design a Turing Machine for accepting the strings of language
    L = $\{ x \in \{a, b\}^* / x=yy$ for some $y \in \{a, b\}^* \}$. Process the string **abab**
43. Design a TM for a set of all strings over {a,b} with equal number of a's and b's. Show the moves of the TM for the input string **ababba.**
44. Design Turing machine to accept all set of palindromes over {0, 1}*. And also write the transition diagram and Instantaneous description on the string 10101
    (or)Design a Turing Machine to recognize any palindromes strings over {0,1}
    (or) Design a Turing Machine to recognize binary palindromes.
45. Construct Turing machine for L = $\{a^nb^m a^{(n+m)} \mid n,m\geq1\}$
46. Design a Turing Machine for L=$\{0^n1^m0^n1^m/m,n\geq1\}$
47. Construct Turing machine for the languages containing the set of all strings of balanced parenthesis?
48. Construct a Turing machine for $L = \{a^ib^jc^k \mid i*j = k;\ i, j, k \geq 1\}$.
49. Design a Turing machines and its transition diagram to accept language greeted by
    $\{a^i b^jc^k/i,j,k \in N, i+k=j\}$.
50. Design a Turing Machine for L = $\{x \in \{a,b\}^*/x$ contains even no. of a's and odd no. of b's$\}$.Show the moves of the TM for the input string **abaabba.**
51. Design a TM for recognizing L=$\{x \in \{a,b\}^*/x$ contains a in the third position from the right end.
52. Design a Turing Machine for recognizing L=$\{x \in \{a,b,c\}^*/x$ contains a or b in the third position from the right end.
53. Design a Turing Machine for recognizing the language $(a+b)^*aba(a+b)^*$. Draw its transition diagram and table. Using the instantaneous description notation process the string **aabaabaaab**
54. Design a Turing Machine for recognizing the language L =$\{x \in \{a,b\}^*/x$ ends with aba$\}$. Specify its transition diagram and table. Process the strings **abaaba** and **ababaa** using ID notation.
55. Construct Turing machine for **L = $\{a^{2n}b^n \mid n\geq1\}$**
56. Construct Turing machine for **L = $\{a^nb^{2n} \mid n\geq1\}$**
57. Construct Turing machine for **L = $\{a^{n+2}b^n \mid n\geq1\}$**
58. Construct Turing machine for **L = $\{a^nb^{n+2} \mid n\geq1\}$**
59. Construct Turing machine for **L = $\{a^nb^{3n} \mid n\geq1\}$**
60. Construct Turing machine for **L = $\{a^{3n}b^n \mid n\geq1\}$**

UNIT-4

1. Define compiler. Describe the logical phases of a compiler with a neat sketch, show the output of each phase, using the example of the following statement
   position := initial + rate * 60
2. Explain the chief functions of lexical analysis phase.
3. What is the role of transition diagrams in the construction of lexical analyzer?
4. How a finite automaton is used to represent tokens and perform lexical analysis with examples.
5. Differentiate between token, lexeme and pattern with examples.
6. Explain the recognition of keywords and identifiers with a suitable transition diagram.
7. Design a non-recursive predictive parser for the following grammar:
   S -> AaAb | BbBb
   A -> e
   B -> e where a, b, e are terminals.
8. Construct LALR parsing table for the following grammar:
   S -> CC
   C -> cC | d
9. Write an algorithm to find LR(0) items and give an example.
10. Define LR(k) parser. Draw and explain the model of LR parser.
11. State and explain the rules used to construct the LR(1) items.
12. Differentiate Top Down parsing and Bottom Up Parsing.
13. Consider the grammar:
    E→ E+E
    E→ E*E
    E→ id
    Perform shift reduce parsing of the input string **"id1+id2+id3".**
14. Construct SLR parsing table for the following grammar:
    S→AS/b
    A→ SA/a
15. Define FIRST and FOLLOW functions. Write the rules for computing FIRST(X) and Follow(X) where X is grammar symbol. Compute FIRST and Follow, for all the non-terminals in the following grammar:
    S→ iEtSS'/a
    S'→ eS/ε
    E→ b
16. Construct CLR parsing table for the following grammar:
    S→ Aa/bAc/bBa
    A →d
    B→ d
17. Define LL(1) grammar. Show that the following grammar is LL(1).
    S→ aABd
    A→ b/ε
    B→ c/ε
18. Construct LL(1) parsing table for the following grammar. Find the moves made by the LL(1) parser on the input string: **id+id*id**
    E → E+T/T
    T → T*F/F
    F→ (E)/id

19. Construct LALR parsing table for the following grammar:
    S → wAz/xBz/wBy/xAy
    A→ r
    B→ r
20. Design LALR(1) parser for the following grammar:
    S -> aAd | bBd | aBc |bAc
    A -> e
    B -> e
    where a, b, c, d, e are terminals.
21. Consider the grammar : **S → (S)/a**
    Construct SLR(1),CLR(1) and LALR(1) parsing tables
22. Show that the following grammar is LL(1).
    S -> AaAb | BbBb
    A -> ε
    B -> ε
23. Find the FIRST and FOLLOW sets of the each of the non-terminals for the following grammar:
    S→ aAB/bA/ε
    A→ aAb/ε
    B→ bB/c
24. Show that following grammar is not LL(1)
    S → i C t S E | a
    E → e S | ε
    C → b
25. What is symbol table? What is its need in compiler design?
26. Explain S-Attributed & L-Attributed definitions in detail

UNIT-5

1. What is role of intermediate Code generator in compilation process? Explain Various Forms Of Intermediate Codes Used By Compiler.
2. Explain various methods of implementing three address statements with suitable examples. (or) Explain about quadruples, triples and indirect triples of three-address statements of intermediate code
3. Write quadruples, triples and indirect triples for the expression:
   -(a*b)+(c+d)-(a+b+c+d)
   (or) For the given expression generate different kinds of three-address codes -(a*b)+(c+d)-(a+b+c+d)
4. Write the short note on: (i) Abstract syntax tree (ii) Polish notation (iii) Three address code
5. Construct abstract syntax tree & DAG for the assignment statement
   x:=a*b+c-a*b+d
6. Write the quadruple, triple, indirect triples for the statement
   a:= b * -c + b * -c
7. Translate the assignment A:= -B*(C+D) into following
   i) Quadruple ii) Triples iii) Indirect triples
8. Translate the expression -(a+b)*(c+d)+(a+b+c) into the following
   i) Quadruples ii) Triples iii) Indirect triples

9. Convert the following arithmetic expressions into Abstract syntax tree, DAG, postfix notation and three-address code:
   i) b*-(a+b)
   ii) a+b*(a+b)+c+d
10. What is code optimization? Compare machine dependent and independent code optimization techniques.
11. What is code optimization? Explain about various levels and types of optimizations
    (or) Explain different principle sources of optimization techniques with suitable examples
12. Discuss about principal sources of optimization.
13. Write short note on
    a. Constant Folding
    b. Dead Code Elimination
    c. Code Motion
    d. Induction Variable Elimination
14. Discuss briefly various loop optimization techniques.
15. Define flow graph. Explain the optimization of Basic Blocks.
16. Write about all issues in code generation. Describe it.
    (or) Explain the different issues in the design of a code generator
17. Explain the peephole optimization Techniques?
    (or) Discuss the transformations that are characteristic of peephole optimizations.
    (or) What kinds of peephole techniques can be used to perform machine-dependent optimizations?+
18. What is a basic block and flow graph? Explain how flow graph can be constructed for a given program.
19. What is peephole optimization? How can it be performed? Give its role in code generation.
20. Discuss about register allocation and assignment in target code generation.