# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

15-Sept

# DISCRETE MATHEMATICS

Kenneth Rosen

9 Marks

Chapters:

1. Logic
2. Combinatorics
3. Set Theory [KOLMAN, BUSAN & ROSS]
4. Graph Theory [NARSINGH DEO]

[Theory]

## LOGIC

1. Logical Statement ? [Proposition]
2. Logical Operators & their properties ($\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$, $\oplus$, $\uparrow$, $\downarrow$) — Disjunction, Conj., Negation
3. Tautology (T), Contradiction (C) & Contingency (CT) [Satisfiable (T or CT) / Unsatisfiable (C)]
4. Normal Forms & properties: PDNF (Principle Disjunctive Normal Form), PCNF (Principle Conjunction Normal Form)
5. Implications & Biconditional ($\Rightarrow$, $\Leftrightarrow$)
6. Arguments & Fallacy [Invalid Argument]
7. Rules of Inference
8. Predicate Logic - Quantifier ($\forall$, $\exists$)
   - Validity of a predicate
   - Properties
   - Translation

# LOGIC

(S, ∘)
↓ ↳ operators
Set of all Logical Stmt

**ogical StateMent** - (Proposition)

- Declarative sentence which can be either true or false but not both.

Ex - This board is white.
This Fan is Rotating [is/will tends to declaration]
- This sentence is true.

***a Logical StateMent**

1. Questions - What is Your Name?
2. Command - Stand up.
3. Exclamation - Oh! That's great.
4. $x+2=4$
   (it is not proposition bcoz for some x value it is true ——— it is false)
5. He is tall. (unless he is specified)
6. Today is Wednesday.
   [Not a proposition bcoz today may be true, but tommorow it will become false]
7. Tommorow it will rain.
   [Not a proposition.]
8. This sentence is false.
   [Negative Self Referential Sentence]

**ogical Operators :**

A proposition is written in the following way:

$P: 2+2=4$

$q(x): x+2=4$ (Predicate) but not a proposition

~~True~~ False — $\forall x\, P(x)$ ] propositions
True — $\exists x\, P(x)$ ]

**gation** - $(\sim, \bar{P}, \neg, p')$ . Unary operators

| P | $\bar{P}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

| P | Negation |
|---|---|
| is | is not |
| isnot | is |
| = | ≠ |
| < | ≥ |
| > | ≤ |
| $p \vee q$ | $p' \wedge q' \Rightarrow p \downarrow q$ |
| $p \wedge q$ | $p' \vee q' \Rightarrow p \uparrow q$ |

| P | Negation |
|---|---|
| $P \Rightarrow q$ | $Pq'$ |
| $p \Leftrightarrow q$ | $p \oplus q$ |
| $p \oplus q$ | $p \Leftrightarrow q$ |
| $p \uparrow q$ | $p \wedge q$ |
| $p \downarrow q$ | $p \vee q$ |

- if $p \vee q = 1$
  than $p = \sim q$ is one possibility but not the sure thing. it also allow some other thing
- if $p \wedge q = 0$
  $\Rightarrow [p = \sim q]$ not allways.
- If $p \vee q = 1$ & $p \wedge q = 0 \Rightarrow \boxed{p = \sim q}$
- If $p \Rightarrow 2+2 = 4$ or $3+7 = 10$
  $\bar{p} \Rightarrow 2+2 \neq 4$ and $3+7 \neq 10$
- If $p \Rightarrow 2+2 = 4$ and $3+7 = 10$
  $\bar{p} \Rightarrow 2+2 \neq 4$ OR $3+7 \neq 10$
- p: 2 is even & divisible by 4.
  p': 2 is odd or not divisible by 4.
- p: if it rains, i will carry umbrella. [Either it does not rain OR I will carry Umbrella]
  p': It rains and I will not carry Umbrella

$\wedge$ - all must be true
$\vee$ - atleast one is true
$\oplus$ - Exactly one is true
$\uparrow$ - Atleast one is false

Conversion of Secondary operators into Basic operators:

- $p \rightarrow q = p' + q$
- $p \Leftrightarrow q = p'q' + pq = (p \oplus q)' = p' \Leftrightarrow q' = p' \oplus q = p \oplus q'$
- ~~$p \Rightarrow p \oplus q$~~ $= pq' + qp' = p' \Leftrightarrow q = p \Leftrightarrow q' = p' \oplus q'$
- $p \Leftrightarrow q = (p' + q)(p + q')$ $[(p \Rightarrow q) \wedge (q \Rightarrow p)]$
- $p \oplus q = p' \oplus q'$
- $p' \oplus q = p \Leftrightarrow q = p \oplus q'$

- p: A number is even if and only if divisible by 2. $[\bar{p} \Rightarrow \bar{q} \wedge p \Leftarrow q]$
  P': A number is even or it is divisible by 2, but not both.

- NOR - Neither ... NOR
- DR - Either ..... OR

Negation for predicate-

| P(x) | $\sim P(x)$ |
|---|---|
| $\forall x\, P(x)$ | $\exists x \sim P(x)$ |
| $\exists x\, P(x)$ | $\forall x \sim P(x)$ |
| $\forall x \sim P(x)$ | $\exists x\, P(x)$ |
| $\exists x (\sim P(x))$ | $\forall x\, P(x)$ |

$\sim(\forall x(P(x)\to Q(x))) \equiv \exists x(\sim[P(x)\to Q(x)])$

$\equiv \exists x(P(x)\wedge\sim Q(x))$

$\sim(\forall x\exists y\, P(x,y)) \equiv \exists x\forall y \sim P(x,y)$

$\sim(\exists x\forall y\forall z(P(x,y,z)\oplus Q(x,y,z))) \equiv \forall x\exists y\exists y(P(x,y,z)\Leftrightarrow Q(x,y,z))$

$p\sim(p\Rightarrow q) = \sim(p'+q)$

| | |
|---|---|
| $p\Rightarrow q$ [stmt] | $(p=0 \text{ or } q=0)\Rightarrow(pq=0)$ |
| $q\Rightarrow p$ [converse] | $(pq\neq 0)\Rightarrow(p=0 \text{ or } q=0)$ |
| $\sim p\Rightarrow\sim q$ [inverse] | $(p\neq 0 \text{ and } q\neq 0)\Rightarrow(pq\neq 0)$ |
| $\sim q\Rightarrow\sim p$ [contrapositive] | $(pq\neq 0)\Rightarrow(p\neq 0 \text{ and } q\neq 0)$ |

inary operators-

| | | p+q | p·q | | | | | |
|---|---|---|---|---|---|---|---|---|
| p | q | $p\vee q$ | $p\wedge q$ | $p\Rightarrow q$ | $p\Leftrightarrow q$ | $p\oplus q$ | $p\uparrow q$ | $p\downarrow q$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

if two prepositions are equivalent (X, Y)

then $[X\Leftrightarrow Y\equiv 1]$ $[X\equiv Y \text{ iff } X\Leftrightarrow Y=1]$

Ex Let $b\Leftrightarrow c$ and $a\Leftrightarrow(b\vee\sim b)$ is tautology
what can be inferred about $a\vee(b\wedge c)$?

$b\Leftrightarrow c \Rightarrow b\equiv c$

$a\Leftrightarrow(b\vee\sim b)\Rightarrow a=1$

$\therefore a\vee(b\wedge c)=a\vee(b\wedge b)=1\vee b=1$ (Tautology)

olean Algebra: $(S,+,\cdot,')$ [Logic, Digital logic, Set theory]

$(S,\vee,\wedge,\sim)$

$(S,\cup,\cap,A^c)$

- No of elements in set of Boolean Algebra must be in power of 2.
- $Q_n$ is a boolean Algebra.

- $a-b=a\wedge b'$
- $A-(B\cup C)=(A-B)\cup(A-C)$ (Test T or F)

$a-(b+c)=(a-b)+(a-c)$

$ab'c'=ab'+ac'$ (false)

## Properties of Operators-

• operators are also known as logical connectives.

1. Closure- $\forall x, y, \begin{bmatrix} x+y \in S \\ x \cdot y \in S \\ \sim x \in S \end{bmatrix}$ OR $\begin{bmatrix} x \vee y \in S \\ x \wedge y \in S \\ \neg p \in S \end{bmatrix}$

$\forall A, B \in S \begin{bmatrix} A \cup B \in S \\ A \cap B \in S \\ A^c \in S \end{bmatrix}$

2. Commutative:

$\forall x, y \in S \begin{bmatrix} x+y = y+x \\ x \cdot y = y \cdot x \end{bmatrix}$ $\forall A, B \in S \begin{bmatrix} (A \cup B) = (B \cup A) \\ (B \cap A) = (A \cap B) \end{bmatrix}$

$\forall x, y \in S \begin{bmatrix} x \wedge y = y \wedge x \\ x \vee y = y \vee x \end{bmatrix}$

3. Associative:

$\forall x, y, z \in S \begin{bmatrix} x+(y+z) = (x+y)+z \\ x \cdot (y \cdot z) = (x \cdot y) \cdot z \end{bmatrix}$

$\forall x, y, z \in S \begin{bmatrix} x \wedge (y \wedge z) = (x \wedge y) \wedge z \\ x \vee (y \vee z) = (x \vee y) \vee z \end{bmatrix}$

$\forall A, B, C \in S \begin{bmatrix} A \cup (B \cup C) = (A \cup B) \cup C \\ (A \cap B) \cap C = A \cap (B \cap C) \end{bmatrix}$

4. Distributive: $\forall x, y, z \in S \begin{bmatrix} x+(y \cdot z) = (x+y)(x+z) \\ x \cdot (y+z) = xy + xz \end{bmatrix}$

$\forall x, y, z \in S \begin{bmatrix} x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \\ x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \end{bmatrix}$

$\forall A, B, C \in S \begin{bmatrix} A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \\ A \cap (B \vee C) = (A \cap B) \cup (A \cap C) \end{bmatrix}$

5. Identity: $\begin{bmatrix} \exists_0 \forall x \; x+0 = x = 0+x \\ \exists_1 \forall x \; x \cdot 1 = x = 1 \cdot x \end{bmatrix}$ & $0 \neq 1$

$\forall x \in S \begin{bmatrix} x \wedge T = x = T \wedge x \\ x \vee F = x = F \vee x \end{bmatrix}$

$\exists \phi \; \forall A \in S \begin{bmatrix} A \cup \phi = A = \phi \cup A \\ A \cap S = A = S \cap A \end{bmatrix}$ [S → Universal set]

Ques- The Smallest Finite Boolean Algebra has $2^n$. What is value of n?

$n = 1 \Rightarrow 2^n \{0, 1\}$ → Greatest Elmt; 0 → Least Element

[+ → Least Upper Bound; · → Greatest Lower Bound]

Complement- $\forall x \exists x' \begin{bmatrix} x + \bar{x} = 1 \\ x \cdot x' = 0 \end{bmatrix}$ $\begin{bmatrix} x \wedge (\sim x) = F \\ x \vee (\sim x) = T \end{bmatrix} \forall x \exists_{\sim x}$

$\forall A \in S \exists A^c \in S \begin{bmatrix} A \cup A^c = S \\ A \cap A^c = \phi \end{bmatrix}$ [S- Universal Set]

Precedence of operators:

[( ) > (') > ∧ > ∨] → for boolean Algebra

• ( ) > ' > · > + > ⇒ > ⇔

Ex- $\left( \left[ \left( p \vee \left( (\sim q) \wedge r \right) \right) \Rightarrow s \right] \Leftrightarrow t \right)$

If an element satisfy complement property, it surely satisfies identity property.

Laws for Boolean Algebra-

Idempotent Law- $\forall p \in S \begin{bmatrix} p + p = p \\ p \cdot p = p \end{bmatrix}$ $\forall p \in S \begin{bmatrix} p \wedge p = p \\ p \vee p = p \end{bmatrix}$

$\forall A \in S \begin{bmatrix} A \cup A = A \\ A \cap A = A \end{bmatrix}$

- The biggest polynomial in Boolean Algebra with degree 1.
- No power, No coefficient exist

Absorption Law

$\forall p \in S \begin{bmatrix} p + pq = p \\ p \cdot (p+q) = p \end{bmatrix}$ $\begin{bmatrix} p + qp = p = p + pq \\ p \cdot (p+q) = p = (p+q) \cdot p \end{bmatrix}$

$\begin{bmatrix} \text{Ex - } pq + pqr + prst + p \\ = pq + pqst + p \\ = pq + p = p \end{bmatrix}$

Ex4 $\begin{bmatrix} p + [(p+q')(q'+s')] \\ \neq p \end{bmatrix}$

Ex $\begin{bmatrix} p \cdot (q + p'r) \\ \neq p \end{bmatrix}$

Ex3 $\left[ p \cdot (q + rs' + p + s' + q'r') = p \right]$

$\forall p \in S \quad \begin{bmatrix} p + p'q = p+q \\ p(p'+q) = pq \end{bmatrix} \qquad \forall p \in S \begin{bmatrix} p' + pq = p'+q \\ p'(p+q) = p'+q \end{bmatrix}$

Ex- $p + q'r + r + p'q$

$= r + p + q$

3) De Morgan's Law : $\forall p,q \in S \begin{bmatrix} (p+q)' = p'q' \\ (p \cdot q)' = p'+q' \end{bmatrix} \begin{bmatrix} \sim(p \vee q) = \sim p \wedge \sim q \\ \sim(p \wedge q) = \sim p \vee \sim q \end{bmatrix}$

Ex- $(p + qr) \Rightarrow s$

Simplified form is $[p + qr]' + s = p' \cdot (q' + r') + s$

for Set Theory- $\forall A, B \in S \begin{bmatrix} (A \cup B)^c = A^c \cap B^c \\ (A \cap B)^c = A^c \cup B^c \end{bmatrix}$

4. Law of Double Complement :

$\forall p \in S \quad (p')' = p$

$\forall A \in S \quad (A^c)^c = p$

- $p' = q$ if and only if $q' = p$

$[p' = q \Leftrightarrow q' = p]$. Always a tautology

5. Domination Law- $\forall x \in S \begin{bmatrix} x + 1 = 1 \\ x \cdot 0 = 0 \end{bmatrix}$

$\forall p \in S \begin{bmatrix} x \vee 1 = 1 \\ x \wedge 0 = 0 \end{bmatrix}$

$\forall A \in S \begin{bmatrix} A \cup U = U \\ A \cap \phi = \phi \end{bmatrix}$ U- Universal Set

Note- 1. $[p' \Rightarrow q] \equiv [q' \Rightarrow p] \equiv (p \vee q)$

(contrapositive set)

# Tautology, Contradiction & Consistency

Ques: Which of the following is tautology?

(a) $p \rightarrow q$ $p \vee q \Rightarrow p \wedge q$

(b) $p \vee q \Rightarrow r \wedge p$

(c) $p \vee q \Rightarrow r \vee s$

(d) $s \Rightarrow s \wedge t$

(e) None

(a) $(p+q) \Rightarrow pq = p'q' + pq = p \Leftrightarrow q$

(b) $(p+q) \Rightarrow rp = p'q' + pr$

(c) $(p+q) \Rightarrow r+s = p'q' + r + s$

(d) $s' + st = s' + t$

**Tautology**- for a preposition, if truth table contain all true values only

OR

$[p \Leftrightarrow 1 \equiv 1]$

- Also known as valid preposition.

Ques- Check whether it is a tautology?

$$\left[\left[(P \Rightarrow q) \wedge (q \Rightarrow r)\right] \Rightarrow (r \Rightarrow p)\right]$$

$$\left[\left[(p'+q)(q'+r)\right] \Rightarrow (r'+p)\right]$$

$$= (p'+q)' + (q'+r)' + r' + p$$

$$= pq' + qr' + r' + p$$

$$= p + r' \Rightarrow \text{Hence it is contigency}$$

Ques- Check whether it is tautology?

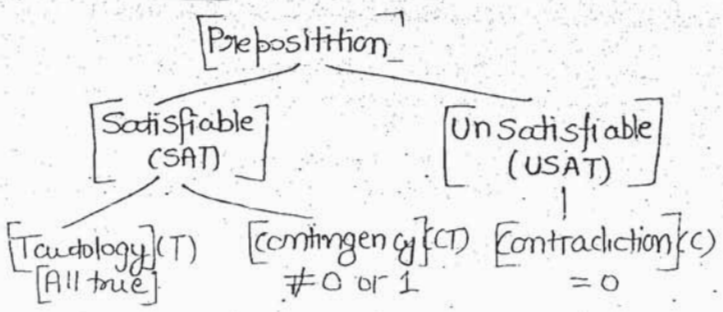
$$\left[\left((p \Rightarrow q) \wedge (q \Rightarrow r)\right) \Rightarrow (p \Rightarrow r)\right]$$

$$= pq' + qr' + p' + r$$

$$= p' + q' + r + q = 1 \quad \text{Hence it is a tautology}$$

**Contradiction**- for all value, it is false $[p \Leftrightarrow 0 \equiv 0]$

- Also known as fallacy.
- Also known as invalid.



# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# ENGINEERING MATHEMATICS.

— Dinesh Sir.

Trapti Singh

## LINEAR ALGEBRA [MATRICES]

- **Properties of Determinant**

1). If 2 rows/columns of a matrix are identical, then their determinant is zero.

$$\Delta = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} = 0$$

2). If 2 rows/columns of a matrix are interchanged, the ~~range~~ sign of determinant is changed.

$$\Delta = \begin{vmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{vmatrix} \quad \text{then} \quad \begin{vmatrix} 3 & 4 & 5 \\ 0 & 1 & 2 \\ 6 & 7 & 8 \end{vmatrix} = -\Delta$$

3). If 3 rows/columns of a matrix are interchanged, then the sign of determinant is unaltered.

$$\Delta = \begin{vmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 0 & 1 & 2 \end{vmatrix}$$

4). In the determinant of a matrix, if any column containing the sum or difference of 2 elements, then it can be split into sum or difference of two determinants.

$$\begin{vmatrix} a & a^2 & a^3+1 \\ b & b^2 & b^3+1 \\ c & c^2 & c^3+1 \end{vmatrix} = \begin{vmatrix} a & a^2 & a^3 \\ b & b^2 & b^3 \\ c & c^2 & c^3 \end{vmatrix} + \begin{vmatrix} a & a^2 & 1 \\ b & b^2 & 1 \\ c & c^2 & 1 \end{vmatrix}$$

5). Determinant of :

$$\boxed{|kA| = k^n |A|}$$

where $k \Rightarrow$ scalar

$A \Rightarrow$ matrix of order $n \times n$

6). $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Longrightarrow \Delta = ad - bc$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Longrightarrow \Delta = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}.$$

7). Lower Triangular Matrix : If all elements above the principal diagonal are 0, then it is said to L·T·M.

$$L\cdot T\cdot M\ A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$$

8). Upper Triangular Matrix : If all elements below the principal diagonal are 0, then it is said to be U·T·M

$$U\cdot T\cdot M. = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

Note: If a Matrix is either L·T·M or U·T·M, then determinant is the product of principal diagonal elements.

$$\Delta(L\cdot T\cdot M) = 1*3*6 = 18$$

$$\Delta(U\cdot T\cdot M) = 1*4*6 = 24$$

Ques. Find the determinant of the matrix.

$$\begin{vmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{vmatrix} = \begin{vmatrix} 0 & a-b & a^2-b^2 \\ 0 & b-c & b^2-c^2 \\ 1 & c & c^2 \end{vmatrix} = (a-b)(b-c)\begin{vmatrix} 0 & 1 & a+b \\ 0 & 1 & b+c \\ 1 & c & c^2 \end{vmatrix}$$

minimum $(n-2)$ operations can be performed only.

$R_1 \to R_1 - R_2$

$R_2 \to R_2 - R_3$

$$= (a-b)(b-c)(b+c-a-b)$$

$$= (a-b)(b-c)(c-a)$$

$$= \begin{vmatrix} 1 & 1 & 1 \\ a & b & c \\ a^2 & b^2 & c^2 \end{vmatrix}$$

Note : $\boxed{|A| = |A^T|}$

**Ques.** Find the determinant of :

$$\begin{vmatrix} 1 & a & b+c \\ 1 & b & c+a \\ 1 & c & a+b \end{vmatrix} = \begin{vmatrix} 1 & a & a+b+c \\ 1 & b & a+b+c \\ 1 & c & a+b+c \end{vmatrix} = (a+b+c)\begin{vmatrix} 1 & a & 1 \\ 1 & b & 1 \\ 1 & c & 1 \end{vmatrix} = \underline{0}.$$

$C_3 \to C_3 + C_2$

**Ques.** Find the determinant of :

$$\begin{vmatrix} \frac{1}{a} & a & bc \\ \frac{1}{b} & b & ca \\ \frac{1}{c} & c & ab \end{vmatrix} = \frac{1}{abc}\begin{vmatrix} bc & a & bc \\ ca & b & ca \\ ab & c & ab \end{vmatrix} = 0$$

**Ques.** Find determinant of :

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1+a & 1 \\ 1 & 1 & 1+b \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ 0 & a & 0 \\ 0 & 0 & b \end{vmatrix} = \underline{ab}.$$

$R_2 - R_1$, $R_3 - R_1$

$$\therefore \begin{vmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 6 \end{vmatrix} = 4 * 5 = \underline{20}.$$

**Ques.** Find the determinant of :

$$\begin{vmatrix} 1+a & 1 & 1 \\ 1 & 1+b & 1 \\ 1 & 1 & 1+c \end{vmatrix} = abc \begin{vmatrix} 1+\frac{1}{a} & \frac{1}{a} & \frac{1}{a} \\ \frac{1}{b} & \frac{1}{b}+1 & \frac{1}{b} \\ \frac{1}{c} & \frac{1}{c} & 1+\frac{1}{c} \end{vmatrix}$$

$$= abc\left(1+\frac{1}{a}+\frac{1}{b}+\frac{1}{c}\right)\begin{vmatrix} 1 & 1 & 1 \\ \frac{1}{b} & \frac{1}{b}+1 & \frac{1}{b} \\ \frac{1}{c} & \frac{1}{c} & 1+\frac{1}{c} \end{vmatrix}$$

$$= abc\left(1+\frac{1}{a}+\frac{1}{b}+\frac{1}{c}\right)\begin{vmatrix} 1 & 0 & 0 \\ \frac{1}{b} & 1 & 0 \\ \frac{1}{c} & 0 & 1 \end{vmatrix}$$

$$= abc\left(1+\frac{1}{a}+\frac{1}{b}+\frac{1}{c}\right) \ldots$$

$$\therefore \begin{vmatrix} 1+a & 1 & 1 & 1 \\ 1 & 1+b & 1 & 1 \\ 1 & 1 & 1+c & 1 \\ 1 & 1 & 1 & 1+d \end{vmatrix} = abcd\left(1+\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}\right) \ldots$$

also. $\begin{vmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{vmatrix} = 5.$

**Ques.** Find the determinant of:

$$\begin{vmatrix} a & a^2 & a^3+1 \\ b & b^2 & b^3+1 \\ c & c^2 & c^3+1 \end{vmatrix} = \begin{vmatrix} a & a^2 & a^3 \\ b & b^2 & b^3 \\ c & c^2 & c^3 \end{vmatrix} + \begin{vmatrix} a & a^2 & 1 \\ b & b^2 & 1 \\ c & c^2 & 1 \end{vmatrix}.$$

$$= abc \begin{vmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{vmatrix} + 1 \begin{vmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{vmatrix}$$

$$= (abc+1)(a-b)(b-c)(c-a).$$

**Ques** If $a, b, c$ are all different and non-zero.

If $\begin{vmatrix} a & a^2 & a^3+1 \\ b & b^2 & b^3+1 \\ c & c^2 & c^3+1 \end{vmatrix} = 0$, then $abc = ?$

here $(abc+1) = 0$

$abc = -1$

**Ques.** Determinant of the matrix is:

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 3 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & -5 & -8 \end{bmatrix}$$

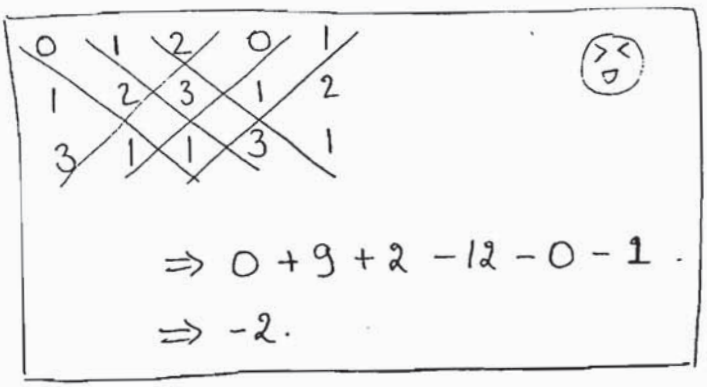
$R_3 - 3R_2$

$\Rightarrow -1[-8+10]$

$\Rightarrow -2.$

[Also by formula. it get the same ans.]

Trick.




$\Rightarrow 0 + 9 + 2 - 12 - 0 - 1.$

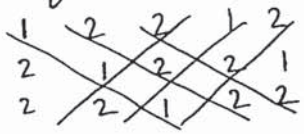
$\Rightarrow -2.$

Jitna bhi slow karo 8 min lagega.

Maggi Taiyaar.

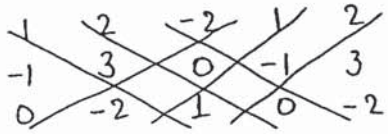
**Ques.** Find the determinant of.

(a) $\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$



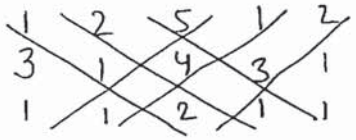
$1 + 8 + 8 - 4 - 4 - 4 \Rightarrow \underline{5}$

(b) $\begin{bmatrix} 1 & 2 & -2 \\ -1 & 3 & 0 \\ 0 & -2 & 1 \end{bmatrix}$



$3 + 0 - 4 + 0 - 0 + 2 \Rightarrow \underline{+1}$

(c) $\begin{bmatrix} 1 & 2 & 5 \\ 3 & 1 & 4 \\ 1 & 1 & 2 \end{bmatrix}$



$2 + 8 + 15 - 5 - 4 - 12 = \underline{4}$.

**Note:** This formula/Trick is applicable on only 3 × 3 Matrix.

## ★ INVERSE OF A MATRIX:

$$A^{-1} = \frac{\text{Adj } A}{\Delta}$$

**Note:⇒** Adj A of 2 × 2 matrix can be find like.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

**Ques.** Find inverse of the Matrix.

(a) $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\quad A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

(b) $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\quad A^{-1} = \frac{1}{4-6} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$

$= -\frac{1}{2} \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$

(c) $B = \begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix}$ $\quad B^{-1} = \frac{1}{6} \begin{bmatrix} 2 & -4 \\ -1 & 5 \end{bmatrix}$

(d) $A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ $\quad A^{-1} = \frac{1}{1} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

(e) $B = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ $\quad B^{-1} = \frac{1}{ab} \begin{bmatrix} b & 0 \\ 0 & a \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix}$

## ⊙ ⇒ Adjacent A for higher order matrix :

(a) **Minor of an element** : The minor of an element in a square matrix is the determinant of the square submatrix in which the row and the column of the particular element lines to be deleted.

For a Square Matrix

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 3 & 1 & 1 \end{bmatrix} \Rightarrow \text{Minor of } 1 = \begin{vmatrix} 0 & 2 \\ 1 & 3 \end{vmatrix} = -2$$

(b) **Cofactor of an element** $= (-1)^{i+j}$ Minor

i.e. $\boxed{\text{cofactor} = (-1)^{i+j} \text{ Minor}}$

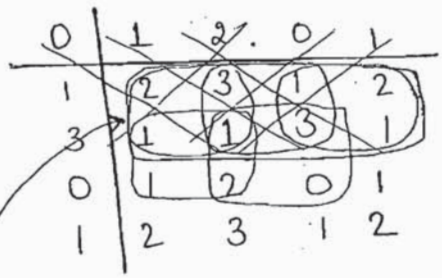
$\therefore$ Cofactor of $1 = (-1)^{3+2}(-2) = 2$.

(c). Adj A → Transpose of co-factor matrix.

TRICK :

Ques. Find inverse of matrix of $3 \times 3$.

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 3 & 1 & 1 \end{bmatrix}$$



$$\Delta = 9 + 2 - 12 - 1$$
$$= -2$$

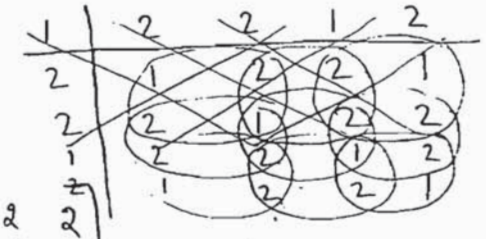
$$A^{-1} = \frac{1}{-2}\begin{bmatrix} -1 & 1 & -1 \\ 8 & -6 & 2 \\ -5 & 3 & -1 \end{bmatrix}$$

Expand horizontally to don't forget to write vertically

Ques. Find inverse of :

(a) $A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$

$\Delta = 1 + 8 + 8 - 4 - 4 - 4 = 5$



$$A^{-1} = \frac{1}{5}\begin{bmatrix} -3 & 2 & 2 \\ 2 & -3 & 2 \\ 2 & 2 & -3 \end{bmatrix}$$

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

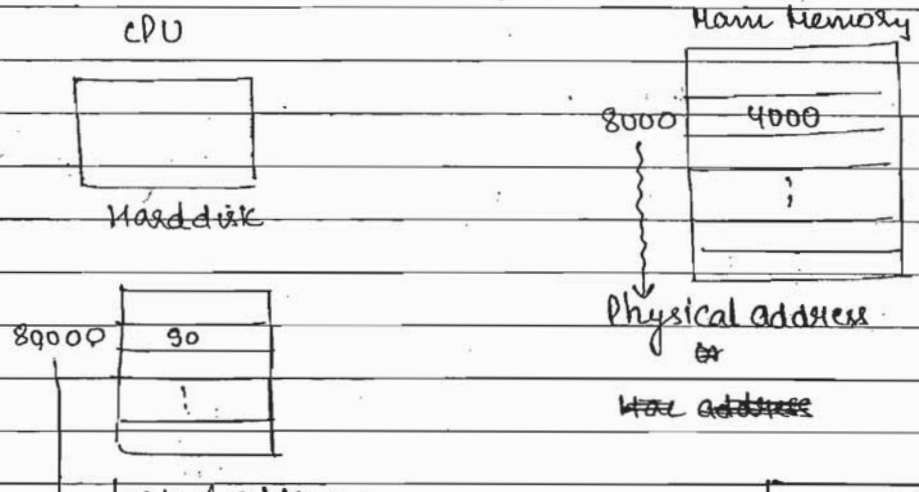
**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# Computer Network and Security




MAC address is local, so cannot be used as unique identification number

Physical address
(or)
Mac address
(or)
~~Mac~~ Ethernet address ⇒ 48 bit address
(or)
LAN card address
(or)
NIC card address

→ Implicit address

① —— ② S.A, D.A — S.MAC, D.MAC — Scope is local

logical address (classful addressing)
32 bit address (IPv4)

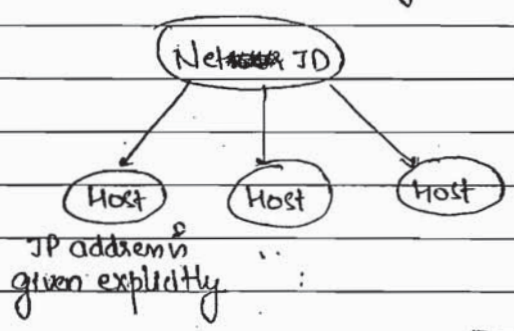
IANA → Internet Assigned Number Authority

Note> Using MAC address alone cannot be used as an identification unit in transmitting the data, because scope is local.

(IP) logical address ⟶ 32 bit address
address
↓ classes ⟶ A, B, C, D, (E) — Research
Unicasting
multicasting

Page. No.
Date : / /

⊙ classful supports two level hierarchy.




www.yahoo.com
hostname

→ Whenever an IP address is assigned to a computer, it is known as host.

→ Entire Network will be represented by a number known as the Net ID.

Notation

i) Binary notation [2]

Ex- 10101111 10101010 10101111 11110000

System friendly. 1st octet 2nd octet 3rd octet 4th octet

ii) Dotted Decimal Notation [10]

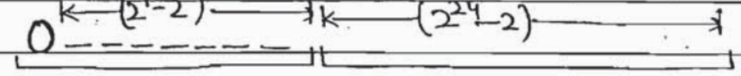
Ex- 143.89.99.126 ——→ User friendly.

——→ In Binary notation starting few bits will decided the type of class

——→ In dotted decimal notation, first octate will decided the type of class.

Page. No.
Date : / /

**Class A>**




Net bits 8bit — Host bits 24 bits

0 0000000 → 0

1111111 → 127

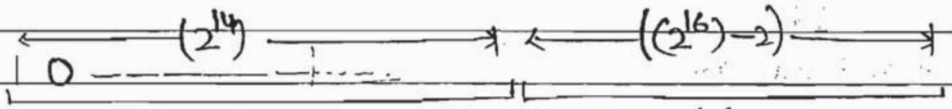
(0—127) but 0 and 127 not used

∴ (1—126) → Class A

0.0.0.0 → DHCP client

127.x.y.z → loop back address

**Class B>**



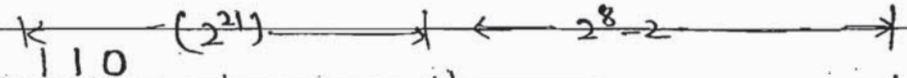

Net bits 16 bits — Host bits 16 bits

1 0 000000 → 128

1 1 1111 → 191

Class B Range → (128—191)

**Class C**




Net bits 24 bits — Host bits 8bits

1 1 0 00000 → 192

1 1 0 11111 → 223

Class C Range → (192—223)

224
5
239

Page. No.
Date: / /

**Class D** — 1110 — no division — multicasting

1110 0000 → 224
1110 1111 → 239

Class D Range → (224–239)

**Class E** — 1111 — Research

1111 0000 → 240
1111 1111 → 255

Class E Range → (240–255)

Q) IP: 201.44.89.99

Net Id =

Direct Broad cast address of network =

Network mask (or Default mask)

Class A: 11111111.00000000.00000000.00000000
mask of A → 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

Mask → allowing and stopping

→ Network mask is a mathematical tool which is used for solving networking problems

IP: 201.44.89.99 (dane)
mask: 255.255.255.0
201.44.89.0 (And)

99: 01100011
0: 00000000
0: 00000000

89: 01011001
255: 11111111
89: 01011001

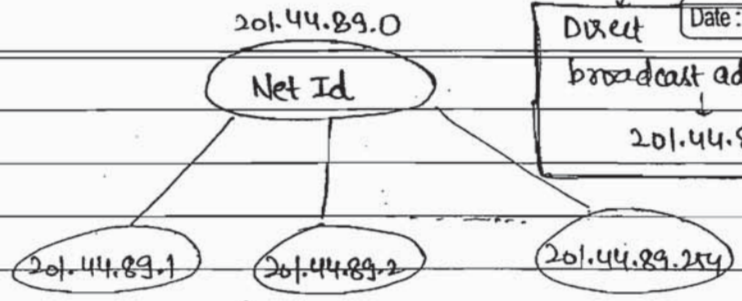
| x | y | output And |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

When want to send address to all the host present

Page No
Date:

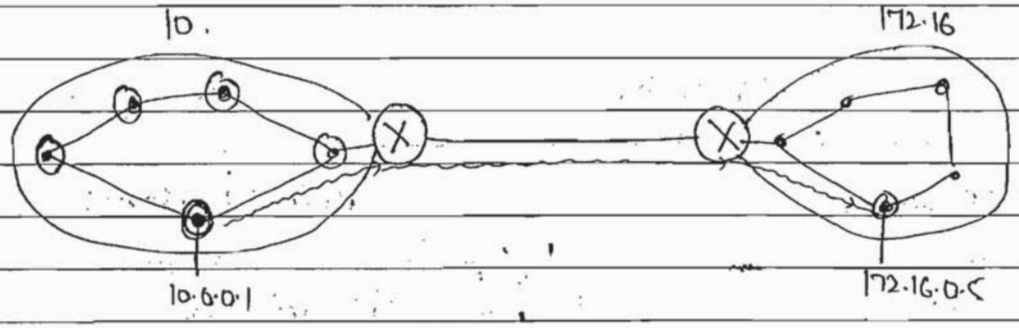
Direct broadcast address.

201.44.89.255




(i) We are subtracting 2 addresses in the no. of host, because one is used for net Id and other one is used for DBA of network

Pseudo approach of network




(i)

| | S.IP | D.IP |
|---|---|---|
| D | 10.0.0.1 | 172.16.0.5 |

ClassB

↓

It is a unicasting packet between the networks (different networks)

(ii)

| | SIP | DIP |
|---|---|---|
| D | 10.0.0.1 | 172.16.255.255 |

all 1's in host bit

Directed broadcast address

| | | |
|---|---|---|
| D | 172.16.255.255 | 10.0.0.1 |

(X) (no meaning)

Signifies Direct broadcast address

→ It is a broadcasting packet on the other network

→ Direct Broadcast address will always be used as Destination IP.

Page. No.
Date : / /

(iii)

| | S.IP | D.IP |
|---|---|---|
| D | 10.0.0.1 | 10.0.0.9 |

It is a unicasting packet within the network. (same network)

(iv) Special case →

| | | |
|---|---|---|
| D | 10.0.0.1 | 255.255.255.255 |

(Broadcast within the network) ↑

↓ limited broadcast address
→ scope is local (LAN)

→ limited Broadcast address will always be used as destination IP.

↓ (used in LAN)

## IP address

### Private IP address

1. Scope is local
2. Work only in LAN
3. By loading networking operating system
4. Ranges of private IP.

| Ranges of private IP | No. of Network |
|---|---|
| 10.0.0.0 — 10.255.255.255 | 1 |
| 172.16.0.0 — 172.31.255.255 | 16 |
| 192.168.0.0 — 192.168.255.255 | 256 |

5. free of cost
6. Will not get Internet service

### Public IP address

1. Scope is globally unique
2. used To get Internet service
3. Not free of cost
4. have control of ISP (Internet Service provider)

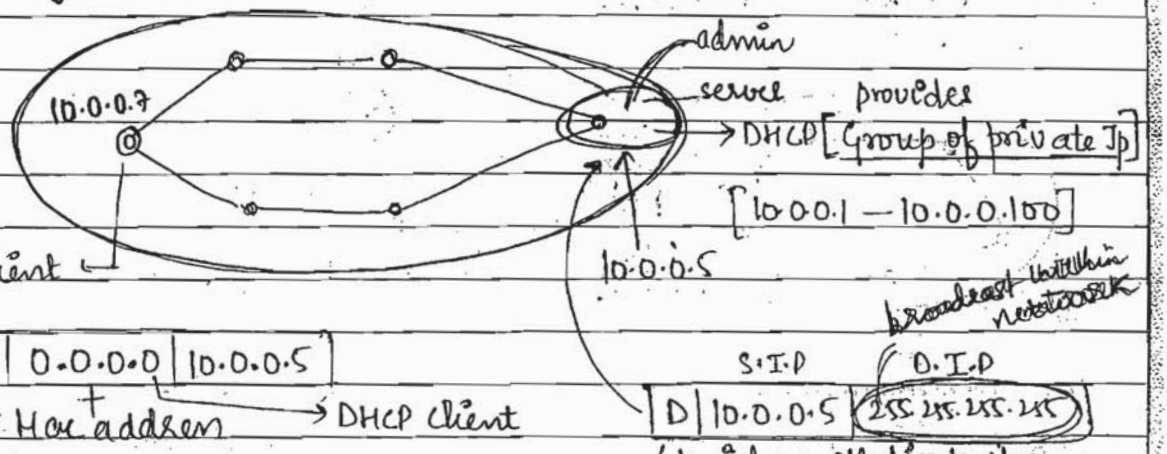
Page. No.
Date: / /

Client ⇒ DOS, XP [DOS commands]

Server ⇒ Window NT, 2003

[DOS commands + Networking protocols (Http, Ftp, Dhcp, —)]

## Assigning Privat IP addresses in LAN (stateful protocol)




Mapping table

| MAC | IP |
|---|---|
| 14:1A:12:1B:14:15 | 10.0.0.7 |
| ⋮ | ⋮ |

① Once the server is loaded with network operating system, it will get group of private IP addresses, out of which one IP is assigned to server

② the servers IP is informed to all the clients using limited Broadcast address.

③ Every client will put a request to the server using DHCP client as the source IP address, along with it Mac address is

# Hindbookcenter

GET IT ON Google Play

ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE

COURIERS SERVICE AVAILABLE

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES

MINIMUM PRICE AVAILABLE @ OUR WEBSITE

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ NOTE;- ALL ENGINEERING BRANCHS

➢ ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE

## PUBLICATIONS BOOKS -

MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK

RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON...OTHERS

HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

Reference: Introduction to Algorithm By coreman

Syllabus: (1) Analysis
- (2) Divide and conquer.
- (3) Greedy Technique
- (4) Dynamic programming.
- (5) Hashing & Tree and graph Traversal.

Definition: It is a combination of sequence of finite steps to solve a problem.

Example: Multiplication of Two Numbers

```
MTN(){ 1. TAKE 2no's (a,b).
       2. Multiply a and b and store result in c.
       3. return c
       }.
```

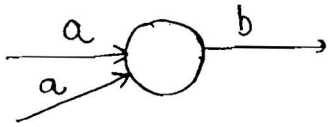
(return c — from which function we have come, we have to return there.)

- finite steps - finite Time should be there (But it doesn't mean finite steps always leads to finite time)
- infine steps - Infinite Time
- All steps are compulsory, so combination is required, so finally it can solve the problem.

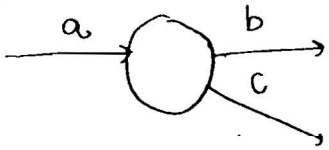
printf ⟶ c
cout ⟶ c++. } syntax

Properties of Algorithm

1. It should Terminate after finite Time.
2. It should produce "atleast" one output (Min^m 1 output)
3. It should take "0 or More input"
4. It should be "deterministic."
   (different behaviour - Non-deterministic)
   deterministic — always same answer.




deterministic (finite steps) also there.




2 O/Ps. Non deterministic.

No dependency → so we can swap the steps of Algo.
Non deterministic → special case.

## Steps Required to Design Algorithm:

1. Problem definition (Knowing problem clearly).
2. Design Algorithm. —— Divide and conquer
   - Greedy Technique
   - Dynamic Prog.
   - Backtracking
   - Branch & Bough (BB).

$\begin{bmatrix} a-b \\ c-d \\ e-f \\ g-3 \end{bmatrix}$

**Algorithm Design:** After knowing the problem, Map the problem to the existing Algorithm.

3. Draw flow chart (Diagramatic Algorithm).
4. Testing and verification. (The Report we made (Test cases), our Prog should Run for those i/Ps)
5. Coding or Implementation.
6. Analysis the Algorithm.

Run – MM (go to Run)
Save – Hard disk
Running Time → MM (space complexity).
time complexity.

} operating system process state diagram.

## Design and Analysis of Algorithm.

C

## Analysis : chapter 1

If your problem having More than 1 solution, Best one will be decided by analysis based on 2 factors.

1. Time complexity (CPU Time)
2. Main Memory (Space complexity).

If your Problem having only 1 soln, go with that soln no no need of Analysis.

**Time complexity:**

Time required for Prog $T(P) =$ compile Time of Prog. $C(P) +$ Running Time of Prog. $R(P)$.

$C(P)$ ⇓ Based on compiler ⇓ S/w ⇓ Based on language of compiler

Based on lang. of Program written compiler is prog.

$R(P)$ ⇒ Based on processor ⇓ H/w ⇓ Type of Hardware.

**Types of Analysis**

1. Apostiary Analysis.
2. Apriori Analysis

postponing the things. (By asking a question, instead of giving answer, asking question to us).

### Apostiary.

① It is based on (dependend) on language of compile & Type of H/w.

Adv
② Exact Answer (It will give exact answer bcoz we are considering Real things).

Dis
③ (constants differ sys to sys) system to system different Answer (diff$^{rt}$ Time)

"it is Relative Analysis".

Hexi processor & compiler lang. is imp.

### Apriori

① It is independent on lang – c. & type of H/w.

② Approximate Answer

Advantage
③ system to system same Answer (same ans with diff$^{rt}$ sys. arch).

"Absolute Analysis".

if prog is Running faster prog. written in great logic.

NOTE: Everyone cannot buy supercomputer but everyone can write super ~~comput~~ algo. because Lord is given same brain to all. But some people will use it some people will not use it.

software company uses – Apriory Analysis.

A Priori Analysis:
we are finding strength of logic.

"It is a determination of order of Magnitude of a statement."
↓
while Running statement is Running how many times.

ex①

```
Main()
{
1. x = y+z;  ⟹ 1 (order of Magnitude).
}
```

(put Big Oh (O) before order of M.

O(1).

ex②

```
main()
{  x = y+3;  ——— 1
   for (i=1; i≤n; i++).
   {
      x=y+3;  ——— n.
   }
}.
```

initialization = 1
condition = n+1
statement = n.
i++ = n.

n+1 = O(n)

Exp③

```
main()
{
   x = y+3;  ——— ①
for (i=1; i≤n; i++)
      x = y+3;  ——— n
for (i=1; i≤n; i++)  ——— n.
{
for (j=1; j ≤ n; j++)  ——— n
   { x = y+3;  ——— n.n
}
}
}.
```

when bracket is in ① statement is their NO bracket.

1 + n(n).
1 + n² = O(n²).

outer loop — add
inner loop = multiply.

Time complexity is finding Bigger loops.
(Where CPU spending More Time).

give this part to cache Memory, so CPU got to know that it is spending More Time, then program is fast.

Locality of Reference — cache Memory; (which is More imp)

Example (4)

```
Main ()
{ while (i≤n)
  {
    i=i+1
    i=i+4
    i= i+5
  }
}
```

increamentation.

$i = i+10 \Rightarrow \frac{n}{10} \Rightarrow \frac{1}{10} \cdot n.$

$\Rightarrow O(n)$.

How many times loop is executing $n/10$.

```
main ()
{ i= n
  while (i≥ 1).
  {
    i= i-1
    i= i-9
  }
}
```

Decrementation.

$i = i-10 \Rightarrow \frac{n}{10} \Rightarrow O(n)$.

*

```
i= i-1  \
i= i-9  /  -10
i= i+7  \
i= i+3  /  10
```

$i = -10+10$

$i = 0$. — not incrementing NO decrementing.

infinite loop

example:5

```
main()
{
 i=1;
while (i≤n).
  {
   i = 2*i
  }
}.
```

$1<64$ ✓
$2<64$ ✓
$4<64$ ✓
$8<64$ ✓
$16<64$ ✓
$32<64$ ✓
$64<64$ X

64 — 6 steps
32 — 5 steps
16 — 4 steps
$n = \log_2 n$.

Proof

| | |
|---|---|
| 1 | 1 |
| 1 | 3 |
| 2 | $3^2$ |
| $2^2$ | : |
| $2^4$ | : |
| : | $3^k = n$ |
| : | $\log_3 3^k = \log_3 n$ |
| | $k = \log_3 n$ |
| $2^k = n$. | |

$\log_2 2^k = \log_2 n$

$k = \log_2 n$

$i = 2*i$
$i = 3*i$

$i = 2*i*3$
$i = 6i$
$k = \log_6 n$

$i = 2*i$
$i = 3*i$
$i = 5*i$ $\Rightarrow$ $i = 30i$
$O(\log_{30} n)$.

II
```
main()
{while (i=n
        i≥1)
   {
    i = i/2 ⇒O(log2n).
   }
}
```

$n$
$n/2$
$n/2^2$
$n/2^3$
:
$n/2^k = 1$
$n = 2^k$
$\log_2 n = k$

$i = i/2$
$i = i/3$
$i = i/4$ $\Rightarrow i/24$.
$\log_{24} n$.

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# Computer Organization

10 marks

Syllabus:

Module 1: computer architecture

Module 2: computer organization.

Ref Books: 1. computer architecture & organigization.
- Morris Mano. (Hardware design)

2. computer orgn.
- william stallings.

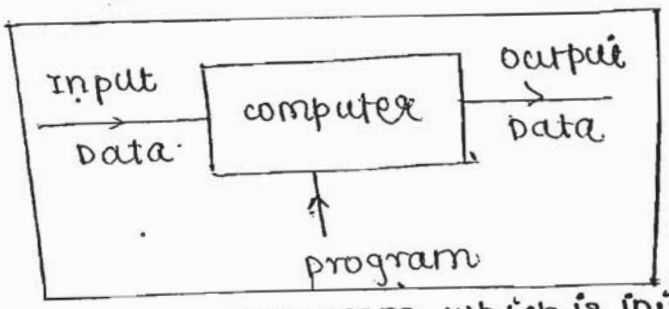
Faculty: Pingili Sagar.

email: sagar262003@yahoo.co.in.

Keywords:

computer:

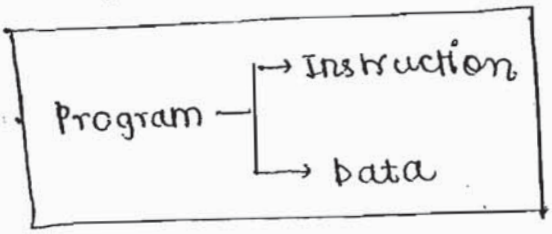
computer is a computational Machine used to process the data under the control of a application program. Therefore computer system functionality is program execution.




(program which is initiated by user)

program:

Program is a sequence of instructions along with the data.




Instruction:

Instruction is a Binary code which is designed inside the processor to confirm some task.

Binary code - Bind - operation with

Eg: If CPU – 'x' supports 8 different operation then opcode = $\log_2 8$ = 3bit.

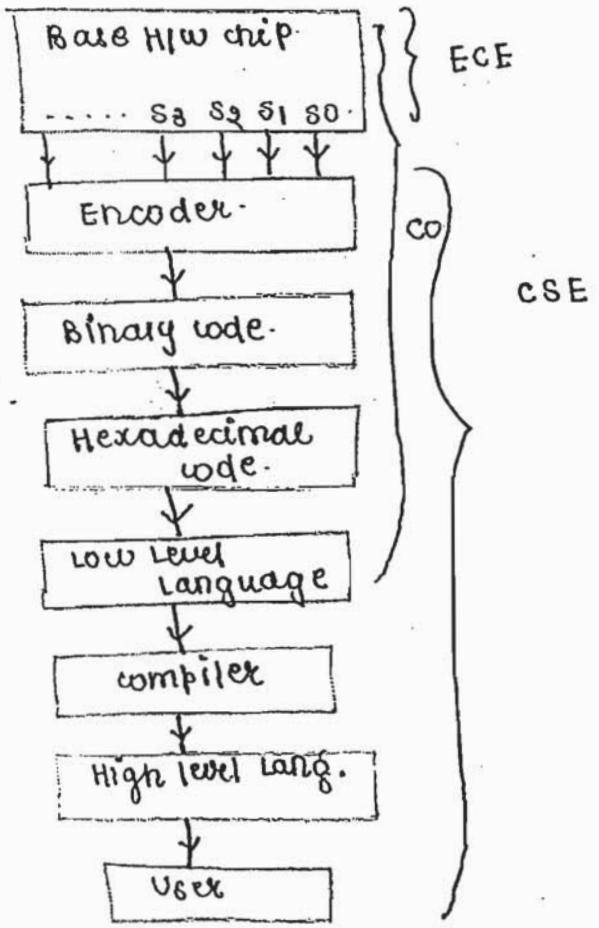
| Binary (opcode) code | operation |
|---|---|
| 000 | + |
| 001 | – |
| 010 | * |
| ⋮ | ⋮ |
| 111 | AND |

Decided by the Designer → ROM → control unit

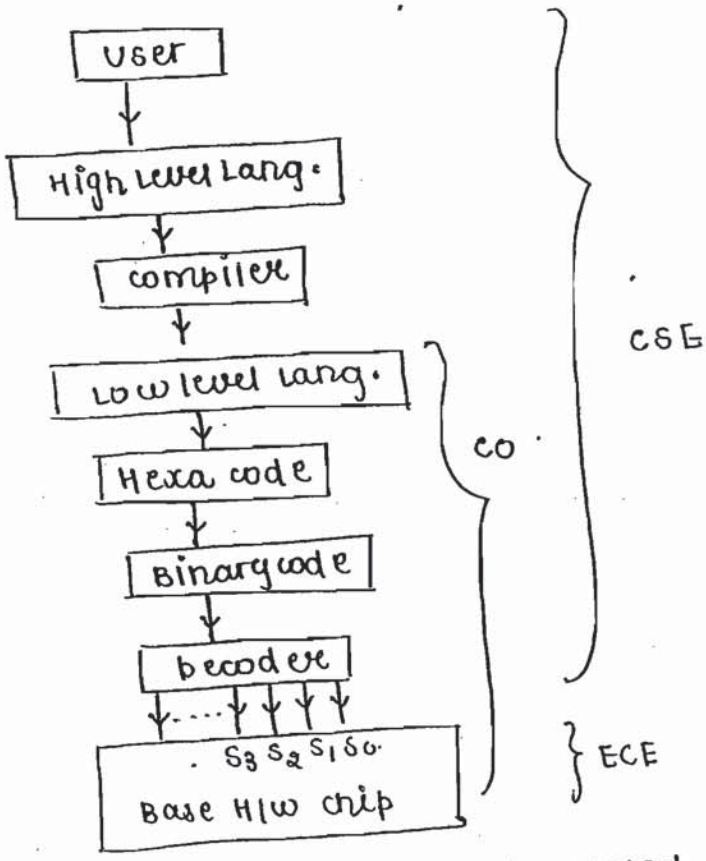
Encoding process: n signals given How many bits required to process signals $\log_2 n$.

Decoding process: n bits are given, How many operation can be performed by computer : '$2^n$' operation.

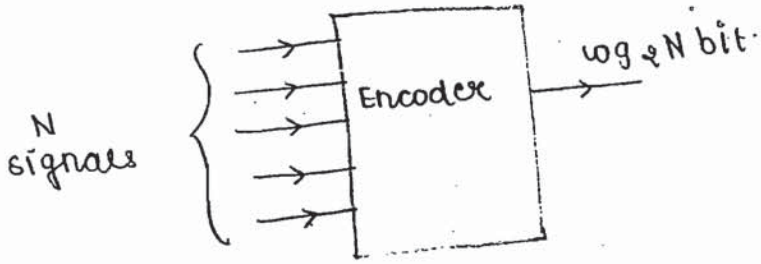
Designer View:



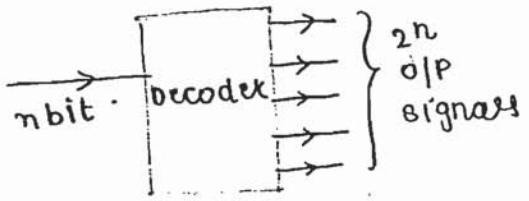
User View :




Encoding: In this process 'N' signals are represented using $\log_2 N$ bit format.



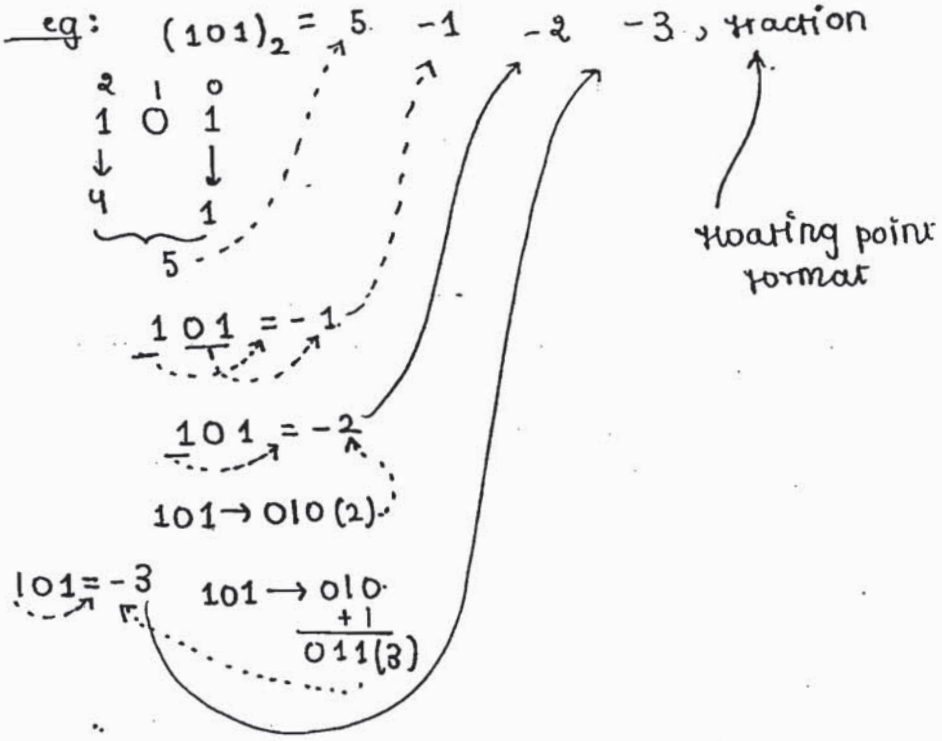

Decoding: In this process, n bit decoder produces $2^n$ output signals.



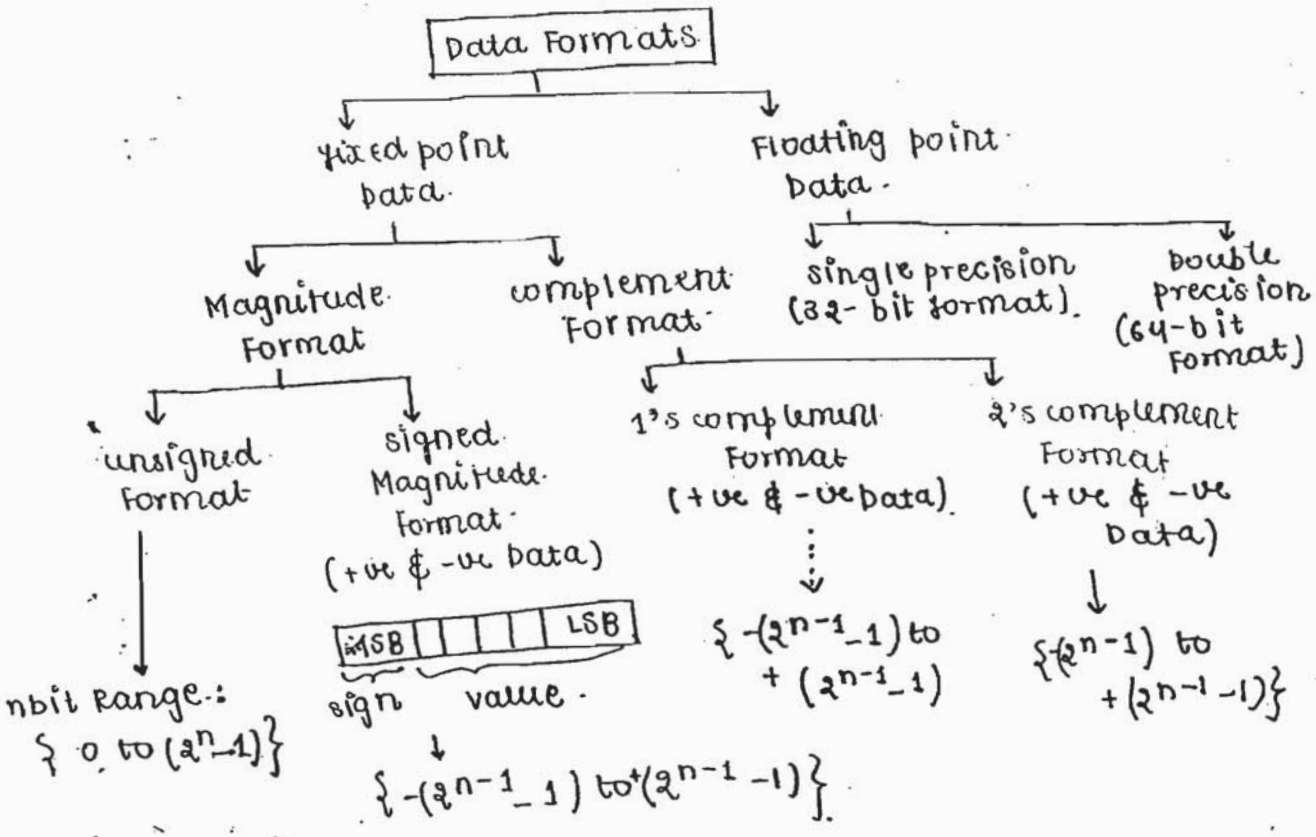
④ Data: It is a Binary code which is associated with a value based on the Data format.

**Binary code - Bind with - value**

eg: $(101)_2$ = 5 -1 -2 -3, fraction




Data Representation:



## Fixed point Data

| 4 Bit Binary. | Unsigned Data. | Sign Magnitude | 1's complement | 2's complement |
|---|---|---|---|---|
| 0000 | 0 | +0 | +0 | +0 |
| 0001 | 1 | +1 | +1 | +1 |
| 0010 | 2 | +2 | +2 | +2 |
| 0011 | 3 | +3 | +3 | +3 |
| 0100 | 4 | +4 | +4 | +4 |
| 0101 | 5 | +5 | +5 | +5 |
| 0110 | 6 | +6 | +6 | +6 |
| 0111 | 7 | +7 | +7 | +7 |
| 1000 | 8 | -0 | -7 | -8 |
| 1001 | 9 | -1 | -6 | -7 |
| 1010 | 10 | -2 | -5 | -6 |
| 1011 | 11 | -3 | -4 | -5 |
| 1100 | 12 | -4 | -3 | -4 |
| 1101 | 13 | -5 | -2 | -3 |
| 1110 | 14 | -6 | -1 | -2 |
| 1111 | 15 | -7 | -0 | -1 |

Data Redundancy Problem. "NOT in USE"

"NOT in USE"

### 1's complument

1000 = -7

111 (7)

1001 = -6

110 (6)

### 2's complument

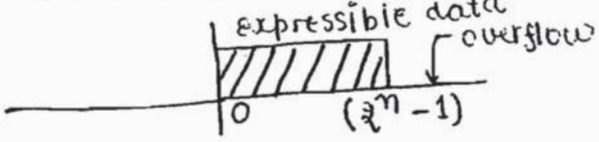
1000 = -8

000
111
+ 1
1000 (8)

1001 = -7

110
+1
111 (7)

unsigned data




eg: 4 bit data {0 to 15}

$$\begin{array}{r} 15 \\ +\,15 \\ \hline 30 \end{array} \qquad \begin{array}{r} \overset{1}{} \\ 1111 \\ 1111 \\ \hline 1110 \end{array}$$

↓ overflow.

Test with 5 bit data : {0 to 31}

NOTE:

(n-bit) + (n bit) = (n+1) bit.

↓

1 bit storage space required.

↓

1 Flip flop

↓

Flag.

↓

carry flag.

condition : "Is there an extra bit out of MSB" → T = set = 1 = C; F = reset = 0 = NC

(or)

"Is Borrow required into the MSB"

eg:

$$\begin{array}{r} 6 \\ \oplus\,7 \\ \hline 13. \\ \hline CY:0 \end{array} \qquad \begin{array}{r} \overset{x\ 1\ 1}{0110} \\ 0111 \\ \hline 1101 \\ \hline CY:0 \end{array}$$

Justify PSW (program status word)

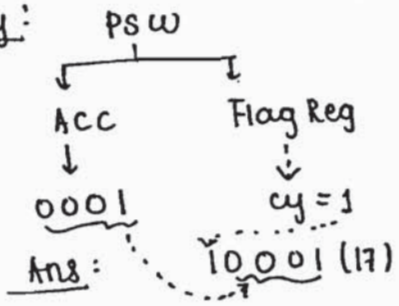
ACC → 1101 - - - - - Flag Register → CY = 0

Ans: 1101 (13)

Eg:

| | |
|---|---|
| 8 | 1000 |
| 9 | 1001 |
| 17 | 0001 |
| cy = 1. | cy = 1 |

Justify:




## Multiplication:

- Multiplicatn process is controlled by a Multiplier.
- Two actions are present in the Multiplication.
  (1) generation of partial product
  (2) summation of partial product
- partial product is generated based on the Multiplier bits. i.e when the multiplier bit is '1' partial product is Multiplicant otherwise partial product is zero. (0).
- After the generation of a partial product, provide the solution to produce the final product.

Multiplicant * Multiplier.

1 1 1 1 * 1 1 1 1 ← LSB

```
      3 1 1 1 1
    3 1 1 1 1 x
  2 1 1 1 1 x x
  1 1 1 1 x x x
-----------------
1 1 1 0 0 0 0 1
```

partial products.

final product

2 → 1 0 → carry (1).

4 → 1 0 0 → carry (2).

6 = 1 1 0 → carry (3)

5 = 1 0 1 → carry (2).

**NOTE:**

(nbit) * (nbit) = 2n bit.

↓

Register pair is used to report the result

**Ques:** consider the following Multiplication.

$(10w1z)_2 * (15)_{10} = (Y01011001)_2$.

what are the value of w, Y, & z variables?

$(15)_{10} = (1111)_2$.

```
      1 0 w 1 z  *  1 1 1 1.
      -----------------------
        2 2 1
      2 1 0 w 1 z.
    2 1 0 w 1 z x
  1 1 0 w 1 z x x
  1 0 w 1 z x x x
  -----------------------------
  1 0 1 0 1 1 0 0 1  ⇒ (Y01011001)2.
  -----------------------------
```

y = 1

(z = 1)

1 + z ⇒ 1 + 1 = 0 (= 1 carry.)

Now replace z with 1.

Now; if (w = 0). for $\boxed{1 + w + 1 + 1 = 0}$

1 + 0 + 1 + 1 = 3 (11) — Not matching

if (w = 1)

1 + 1 + 1 + 1 = 4 (100) — matching.

↳ carry (2)

(w = 1)

**NOTE:** In a Manual Multiplication process, 2 limitations present

(1) Requires More Registers to Hold the partial product

(2) summation process become complex in the H/w Therefore optimization Required that is Accumulated addition.

described in Flow chart

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# COMPILER

Grammar $G = (V, T, P, S)$ — $S$: Start Symbol; $V$: Variable; $P$: Production; $T$: Terminals

Example → $S \longrightarrow ABC$

$AB \longrightarrow CD$

$C \longrightarrow a$

$D \longrightarrow b$

variables : {S, A, B, a} — given by grammar (check), otherwise if not given, consider Capital letters as variables.

## Types of Grammar (according to Noam Chomsky)

1. Type–0 (unrestricted Grammar) — By default, every grammar is Type–0.
2. Type–1 (Context Sensitive Grammar)
3. Type–2 (Context Free Grammar)
4. Type–3 (Regular Grammar)

Type–0 → $\alpha \longrightarrow \beta$ — unrestricted because no restrictions

where $\alpha, \beta \in (V+T)^*$

Type–1 → ① Type–0 ( $A \to \epsilon$, production not allowed)

② $|\alpha| \leq |\beta|$

Type–2 → ① $A \longrightarrow \beta$ (single variable on left side) no restriction on right side.

② $A \in V$

$\beta \in (V+T)^*$

Type–3 → ① $A \longrightarrow \beta T^* / T^*$ (Left linear Grammar)

or

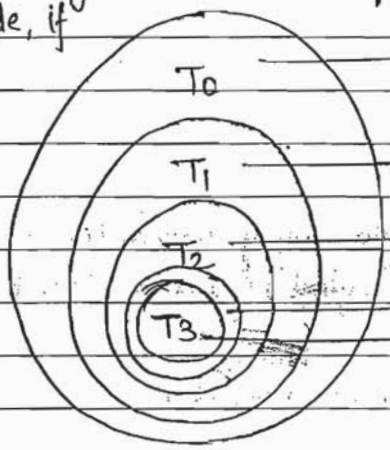
$A \longrightarrow T^* \beta / T^*$ (Right linear Grammar)

Page. No.
Date : / /

Ex → S → ABC
A → ab
B → CD

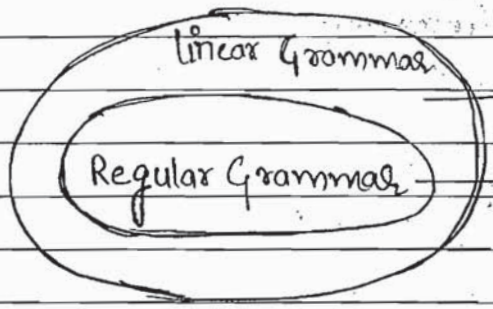
✓ $T_2$, ✗ $T_3$ (Right side)
CFG

① First check left side, single variable, $T_2$ confirmed
② Then check Right side, if doesn't flow left and Right linear, then not $T_3$ (X)




Unrestricted grammar
context sensitive grammar
context free grammar
Linear Grammar
Regular grammar

$V \rightarrow T^*VT^* / T^*$ ⟶ Linear Grammar (In between Type2 and Type3)




contains both left and right linear
only one at a time

## Context Free Grammar (CFL)

① Write a context free Grammar for a language
$L = \{a^m b^n \mid m, n \geq 0\}$

$S \rightarrow AB$
$A \rightarrow aA \mid \epsilon$
$B \rightarrow bB \mid \epsilon$

Page. No.
Date : / /

$\epsilon$ ⟶ string of length 0 (empty string)

$L = \{\}$ ⟶ Empty language

ϵ-DFA

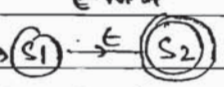
$L = \{\epsilon\}$ ⟶ 


language containing string of length 0

$L = \{\}$ ⟶ 


Empty language (no final state)

ϵ-NFA

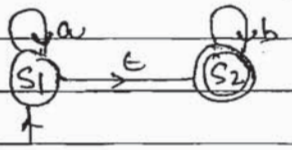



Transition function → DFA : $Q \times \Sigma \longrightarrow Q$ (any state × any input ⟶ goes to one of state)

NFA : $Q \times \Sigma \longrightarrow 2^Q$

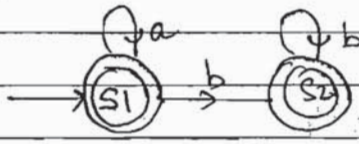
ϵ-NFA : $Q \times \Sigma \cup \{\epsilon\} \longrightarrow 2^Q$ (can go to any number of states; includes ϵ)

$a^*b^*$ ⟶ (Valid-Y, ϵ)



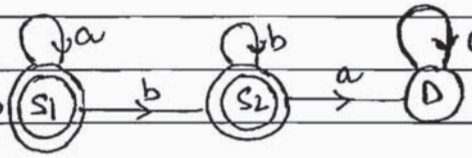

(ϵ-NFA)

(Valid-Y)




(NFA)

(Valid-Y, Invalid-N)




(DFA)

| |
|---|
| Dead or Trap state → Permanent Non-final states. |
| Non-final state ⟶ Temporary Non-final states |

→ Can DFA have more than one final state?

↳ DFA can have multiple final states and dfa doesn't accept the NULL move (ϵ-X dfa)

Page. No.
Date : / /

② Give context free Grammar for Language

$L = \{ a^n b^n c^n \mid n, n \geq 1 \}$

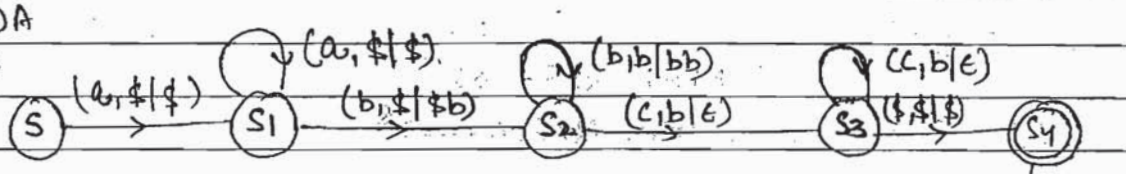
$S \longrightarrow AB$

$A \longrightarrow aA \mid a$

$B \longrightarrow bBC \mid bc$

PDA = Finite Automata + 1 Stack

DPDA




PDA → acceptance by final state
→ acceptance by empty stack

acceptance by final state

③ Give CFG for the language

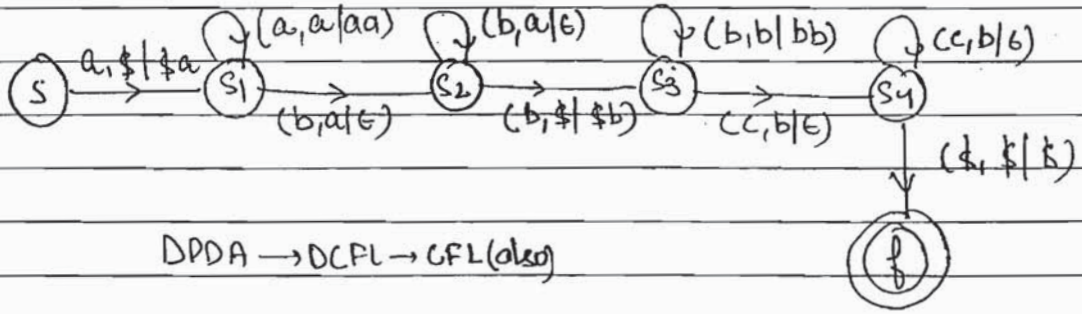
$L = \{ a^i b^{i+j} c^j \mid i, j \geq 1 \}$

$a^i b^i b^j c^j$

$S \longrightarrow AB$

$A \longrightarrow aAb \mid ab$

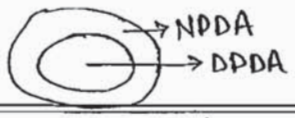
$B \longrightarrow bBC \mid bc$




DPDA → DCFL → CFL (also)

→ NFA is equivalent to DFA

→ NPDA has more power than DPDA.

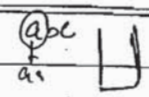



aabbcc


Page. No.

Date: / /

(4) $L = \{ a^m b^n c^n \cup a^m b^n c^n \mid m, n \geq 1 \}$




$S \longrightarrow S_1 / S_2$

$S_1 \longrightarrow AB$

$A \longrightarrow aAb / ab$

$B \longrightarrow cB / c$

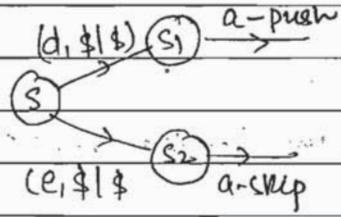
$S_2 \longrightarrow CD$

$C \longrightarrow aC / a$

$D \longrightarrow bDc / bc$

NPDA → CFL language

$L = \{ d a^m b^m c^n \cup e a^m b^n c^n \mid m, n \geq 1 \}$




→ no ambiguity

→ DPDA

(5) CFL for language, $L = \{ a^n b^n c^n \mid n \geq 1 \}$

not CFL → It is CSL

LBA = Finite Automata + 2 stacks

(1) Union of two DCFL's need not be DCFL.

Ex → (4) above. $L = \{ a^m b^m c^n \cup a^m b^n c^n \mid m, n \geq 1 \}$

DCFL CCFL

Union

CFL (here in this example)

But can be possible with some example or not.

⊙ Intersection of two DCFL's, need not be DCFL.

⊙ Intersection of two CFL's, need not be CFL.

⊙ CFL are not closed under complementation.

Ex → $a^n b^n c^n$ ← CSL — complement → Other than $a^n b^n c^n$ comb — CFL

(Compiler part)

⑥ Given CFG for language L = set of all Arithmetic expressions over the $\alpha, \beta$, id.

$E \rightarrow id \mid E+E \mid E*E \mid E-E \mid E/E \mid (E)$

$id \rightarrow +/-/*/\,/$

$\alpha \rightarrow (id+id)\alpha$

$A\alpha \rightarrow id\alpha/id\beta A$

$\alpha \rightarrow +/-/*/$

⑦ Give CFG for language
L = set of all Boolean expressions, over the alphabet 0 and 1.

$B \rightarrow 0/1/B \text{ or } B/B \text{ and } B/\text{not } B/(B)$

⑧ Give CFG for language L = set of all Regular Expressions over the alphabet a, b ($\Sigma = \{a, b\}$).

$R \rightarrow \epsilon/a/b/R^*/R+R/R*R/(R)$

$(a+b)^*$

$A \rightarrow a/b/A^*/(A+A)$

$\epsilon$

$a^* b^*$

$R*$

$(R+R)^*$

$(a+b)^*$

$a(a+b)^*$

$R\cdot(R+R)^*$

$a\cdot(a+b)^*$

⑨ Consider the following Grammar

$S \rightarrow as/sa/a$

i/p string : aaa

How many parse tree?
(Derivation tree or syntax tree)

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

→ Revision
2 (2-3 days)
→ short notes

Page. No.
Date: / /

DBMS

| | More Practice | Less Practice |
|---|---|---|
| Normalization | • Finding candidate key<br>① Lossless Join / DP<br>② INF to BCNF def / theory<br>③ Higher NF? | ① Canonical cover<br>② Decomposition into higher NF. |
| Queries | ① RA / SQL | ② TRC (basics) |
| File & indexing Organisation | ① B / B+ tree [Theory / Numerical] | ② Dense index / Sparse index. |
| Transaction and concurrency | ① CSS / VSS | ② ACID<br>Re / CLR / St<br>lock<br>TSO |
| ERD | ① Min RDBMS table for ERD | |

DM and BA and Apti

Algo + CFDS

DBMS — (8–10)
TOC — (8–10)
CD — (3–5)
Digital — (3–5)

90%
(Parser + SDT)

60-70
DFA/NFA
min etc

20-30%
PDA
TM

| CN | CO | OS |
|---|---|---|
| (10 mins) | (8–10) | (8–10) |

① Top down Approach (Kruze)
② Problem

DBMS [8–10 Marks]

Syllabus →
1. Integrity Constraints and ER-model — 2 Marks
2. Normalization. — 2 Marks
3. Queries [Relational Algebra, SQL, Relation calculus] — 4 Marks
4. File Organisation and Indexing. — 2–4 Marks
5. Transactions and concurrency control. — 2–4 Marks

→ Notes (Revise 2 times a day)

→ WOB / GATE

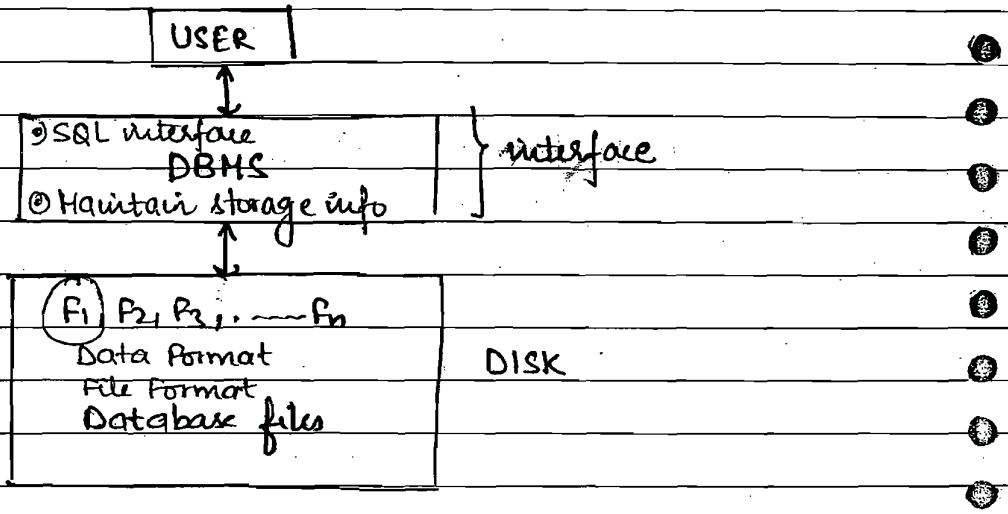
→ Text book Ex Problems
  - 1) DBMS by Raghuramakrishan
  - 2) DBMS by Navathe

# Introduction to DBMS

Page. No.
Date : / /

- Database → Collection of related data
  Ex → set of student's information

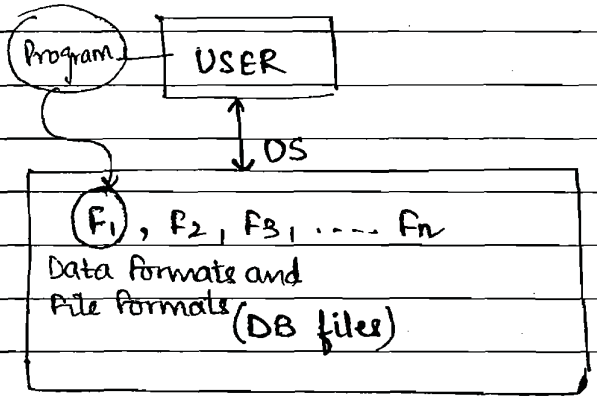
- Database Management System (DBMS) → Software used to manage and access database more efficient manner. In other words, acts as interface between user and database files




User is

Data Independency → User can access data from database files without knowing storage information of file system. In other words, hidding storage information to the external users, is called data independency

Flat File System [DS files] → ~~when the user ma~~
Database files managed by user without DBMS software.




- Flat File System can be used to manage small database.
- Flat file System fails to manage, if size of database is too huge.

## Limitations of Flat File System Vs Advantage of DBMS file System

| Limitation of Flat File System | Advantage of DBMS file system |
|---|---|
| 1) Too Complex to manage and develop application programs | 1) Because of data independency, easy to develop application programs (simple SQL Query required to access the data). |
| 2) More I/O cost to access data from database files | 2) Because of Indexing, less I/O cost required to access data from database files. |
| 3) less degree of concurrency. | 3) More degree of concurrency. |
| 4) Too complex to maintain, ~~non~~ non-redundant data. | 4) By using normalization of database, can maintain non-redundant data. |

Page. No.
Date : / /

# Integrity Constraints

Data Models

- **RDBMS** → Relational DBMS (RDMS)
  - → Proposed by R.T. Codd [Codd's data Model]
  - → Codd proposed 12 Rules for RDMS software design (RDBMS Guidelines)
- ODBMS — Object database management system
- NWDBMS — Network database management system
- Hierarchical DBMS

## RDMS Guidelines

1. Data in DB file must be in tabular format [is set of Rows and Columns].
2. No two rows of DB table should be same. [candidate key]

let candidate key — each column is called Attribute / Field

Relational instance — Stud

| Sid | Sname | DOB |
|---|---|---|
| S1 | A | 2000 |
| S2 | B | 1995 |
| S3 | B | 1998 |
| S4 | C | 1995 |

← each row is called Record / tuple

RDBMS · RDBMS file

(Arity for above)

- Arity → No. of attributes of database table. (Ex- Sid, Sname, DoB - 3)
- Cardinality → No. of Records of database table (above → 4)
- Relational Instance (Snapshot) → A record set of database table.
- Relational Schema → Defination or structure of database table.

Ex → Stud(Sid, Sname, DOB) → Relational Schema.

Key for a Relation is defined according to the requirements of user. Key can be one, which can uniquely distinguish it from others.

Page. No.
Date : / /

Candidate Key → Minimal set of attributes used to differentiate records of a relational schema, uniquely.

| | |
|---|---|
| minimal Attribute set | + No two records with same attribute value (unique) — Candidate Key |

Ex → (1) Stud(Sid, Sname, DOB)

Sid : Candidate key.

Sid Sname : Not Candidate Key

(2) Enroll(Sid Cid Fee)

| | | |
|---|---|---|
| S1 | C1 | — |
| S1 | C2 | — |
| S2 | C1 | — |
| S3 | C2 | — |

Sid Cid : Candidate key.

Ex → A student having student id can enroll in more than one course and a course can be enrolled by more than one student.

Primary key can be → student-id (here unique)

→ More than one candidate key is allowed

(3) Emp

| eid | ename (Not NULL) | DOB | PanID | Aadhaar | IFSC | Acno |
|---|---|---|---|---|---|---|
| e1 | A | | X2 | | SB101 | 101 |
| e2 | B | | NULL | | SB101 | 102 |
| e3 | A | | X5 | | IC101 | 101 |
| e4 | C | | NULL | | IC101 | 102 |

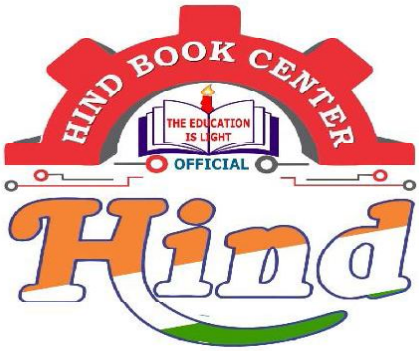
→ NULL : Unknown / unexisted value

→ Not Null constraint : Null values not allowed

Candidate key of Emp relation → {eid, PanID, Aadhaar, ifsc Acno}

eid → Primary Key (Keyword used [PRIMARY KEY])

PanID, Aadhaar, ifsc Acno → Alternative Keys [UNIQUE] - Keyword used




# Hindbookcenter

GET IT ON Google Play

# Hind Book Center & Photostat

## MADE EASY

### Computer Science Engineering / IT
### Toppers Handwritten Notes
### Digital Logic
**By- Sriniwas Sir**

- Colour Print Out
- Blackinwhite Print Out
- Spiral Binding,& Hard Binding
- Test Paper For IES GATE PSUs IAS, CAT
- All Notes Available & All Book Availabie
- Best Quaity Handwritten Classroom Notes & Study Materials
- IES GATE PSUs IAS CAT Other Competitive/Entrence Exams

**Visit us:-www.hindbookcenter.com**

**Courier Facility All Over India**
**(DTDC & INDIA POST)**
**Mob-9711475393**

# Hindbookcenter

GET IT ON Google Play

ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE

COURIERS SERVICE AVAILABLE

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES

MINIMUM PRICE AVAILABLE @ OUR WEBSITE

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ NOTE;- ALL ENGINEERING BRANCHS

➢ ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE

## PUBLICATIONS BOOKS -

MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK

RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON...OTHERS

HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

[5-8 Marks]

# Digital Logic Design

Sri Nivas Sir.

## Syllabus:

Text Book
Modern Digital Electronics
- RP Jain
TMH Publication.

SRINIVAS BETHI
9959750099

1). Boolean Algebra
- Boolean Variables
- Boolean operators
- Logic Gates
- B.A. properties
- Derived operators
- Universal Logic Gates
- Boolean function using Universal logic Gates
- Simplification
- Self Dual functions etc.

2). Number System
- Conversion $(\ )_{r_1} = (\ )_{r_2}$
- $(r-1)$'s and $r$'s
- Signed B. No. System etc.

3). K-Map
- Implicant
- Prime implicant
- Essential Prime implicant.

4). Combinational Circuits
- Code converters
- Arithmatic Circuits
- MUX, Decoder,
- Encoder, DeMUX etc.

5). Sequential Circuits
- Binary Latch
- Flip-flops
- Flip Flop Conversions
  - FF1 $\longrightarrow$ FF2
  - $x_1 y_1 \longrightarrow x_2 y_2$
- Registers $\Rightarrow$ SISO, SIPO etc..

6). Counters
- a) Asynch.
- b) Synch.

## Boolean Algebra ⟷ Chapter 1

$r=2$ ← Binary.

~~Binary~~

Boolean Variables ⇒ A, B, C ...
a, b, c ...

Operators ⇒ OR, AND (Binary), NOT (Unary).

**OR Operator** → +, ∪, ∨

$Y = A+B = A \cup B = A \vee B$

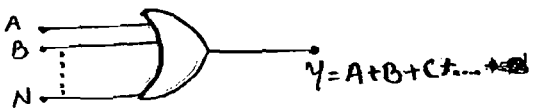
$Y = A+B+C+\dots\dots$

| A | B | Y = A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| A B C | Y = A+B+C |
|---|---|
| 0 0 0 | 0 |
| 0 0 1 | 1 |
| 0 1 0 | 1 |
| 0 1 1 | 1 |
| 1 0 0 | 1 |
| 1 0 1 | 1 |
| 1 1 0 | 1 |
| 1 1 1 | 1 |

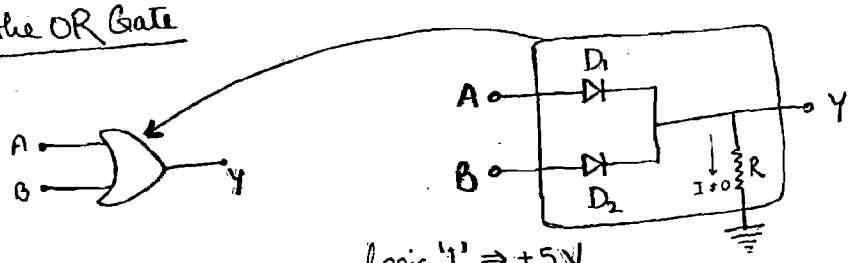
**Note:** • The result of OR operation is zero if and only if, all the variables are zero.

• OR Gate




• No. of inputs in the logic gate is known as Fanin of the logic gate.

For 2 Fanin of the OR Gate




logic '1' ⇒ +5V
logic '0' ⇒ 0V

Truth Table is one consisting of all possible combination of the variables along with the result.

for $n \Rightarrow 2^n$ Rows

values $\Rightarrow [0, 1, 2 \ldots\ldots (2^n - 1)]$

AND OPERATOR : $\longrightarrow \cdot , \cap , \wedge$

$$Y = A \cdot B = A \cap B = A \wedge B$$

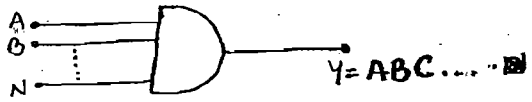
$$Y = A \cdot B \cdot C \ldots\ldots$$

| A | B | $Y = A \cdot B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | B | C | $Y = A \cdot B \cdot C$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Note : • The result of AND operation is zero, if atleast one of the variable is zero.

• AND Gate




#

| | OR | AND |
|---|---|---|
| if $A = 0 \Rightarrow Y =$ | B | 0 |
| if $A = 1 \Rightarrow Y =$ | 1 | B |
| if $A = B = x \Rightarrow Y =$ | x | x |
| if $A \neq B \Rightarrow Y =$ | 1 | 0 |
| Enable input $\Rightarrow$ | 0 | 1 |
| Disable input $\Rightarrow$ | 1 | 0 |

exp.
In OR Gate
$Y = A + B + C$
if $B = 0$, $Y = A + C$
if $B = 1$, $Y = 1$

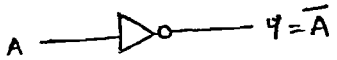
• Enable i/p is the one, it makes the device active.
Disable i/p is the one, it will make the device is to be inactive.

## NOT Operator : '—' , ' ' ;

$Y = \overline{A} = A'$

| A | Y = NOT A |
|---|---|
| 0 | 1 |
| 1 | 0 |

**Note :** • NOT Gate




• NOT operator is also known as Inverter.

## BOOLEAN ALGEBRA PROPERTIES :

1). $A + A + \dots = A$
$A \cdot A \cdot A \cdot \dots = A$

2). $A + 0 = A$
$A \cdot 1 = A$

3). $A + 1 = 1$
$A \cdot 0 = 0$

4). $A \cdot \overline{A} = 0$
$A + \overline{A} = 1$

5). $\overline{\overline{A}} = A$

6). $A + BC = (A+B)(A+C)$ Distributivity.
$A \cdot [B+C] = AB + AC$ Dual of (6).

7). $A + \overline{A}B = A + B$
$A[\overline{A} + B] = AB$

8). $\overline{A} + AB = \overline{A} + B$
$\overline{A}[A+B] = \overline{A}B$

9). $A + AB = A$
$A[A+B] = A$

• **Dual Operation** → OR ↔ AND, 0 ↔ 1.
**Principle of Duality**

**Ques.** $\overline{A} + B[C + \overline{D}(\overline{E} + F)]$

Dual ⇒ $\overline{A}[B + C[\overline{D} + \overline{E}F]]$

• Variable 'x' ⇒ 'x'
Literal 'x' ⇒ 'x' or '$\overline{x}$'

• There is NO effect of the dual operation on the literal.

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# Operating Systems

Tripti Singh..

## Teaching Schedule

I. Introduction and Background.

II. Process Management
- → process concept
- → CPU scheduling
- → Synchronization
- → Concurrent Programming.
- → Deadlocks
- → Threads.

III. Memory Management.
- → RAM Chip Implementation
- → Loading, Linking & Address Binding
- → Techniques
  - paging
  - Multilevel paging.
  - Inverted paging
  - Segmentation
  - Segmented Paging.
- → Virtual Memory.

IV. File Systems.

## Textbooks

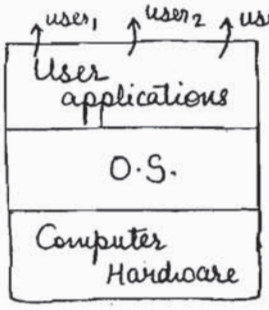
1. OS by Galvin.
2. Modern OS by A.S. Tenenbaum.
3. OS by William Stallings.

# Chapter 1

## Introduction and Background

Q. What is an OS?

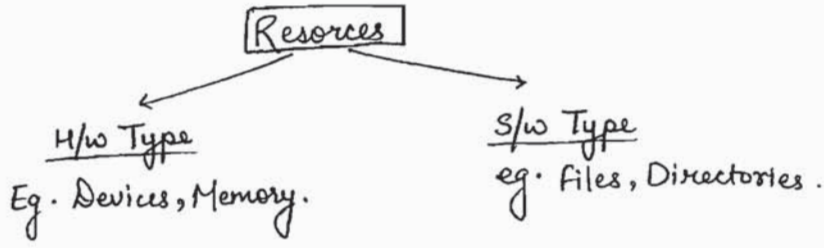
OS is an interface between user and computer hardware.




```
main()
{
  int x;
  printf("Hello");
}
```

→ internally calls write() System Call inorder to communicate with the monitor.

- System Call: System call is the request made by the user program to the OS in order to get any kind of service.
- Operating System is also called as Resorce Allocator because it is responsible for allocating resorces of a computer.




Goals of O.S.

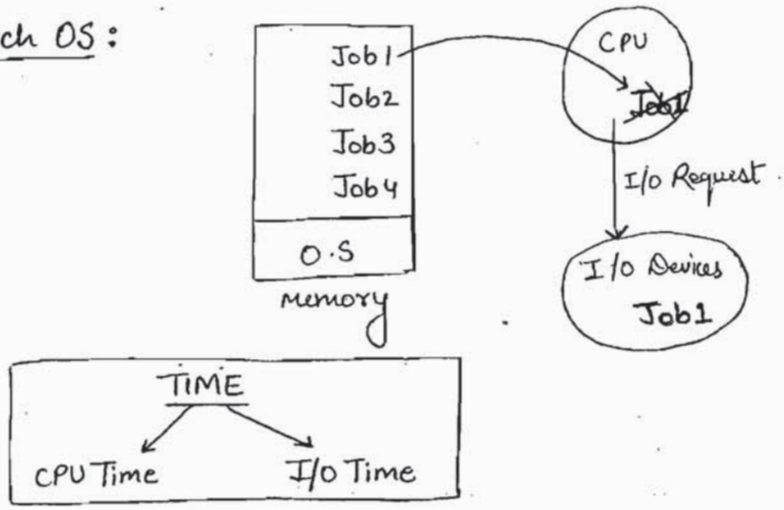
1. The primary goal is convienience. (easy to use)
2. The secondary goal is efficiency. (Stability).

~~Types of OS~~

~~2).~~

## Types of OS

### (1). The Batch OS:

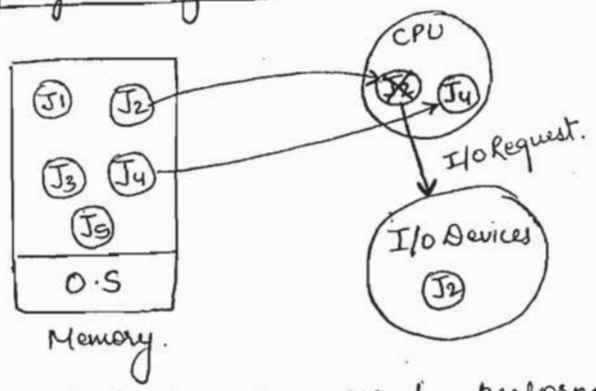



- If the Job is completed completely then only another Job will be scheduled onto CPU.
- Increased CPU idleness.
- Decreased throughput of the system.

**Throughput**: No. of jobs completed per unit time is called throughput of the system.

**Exp**: IBM OS/2

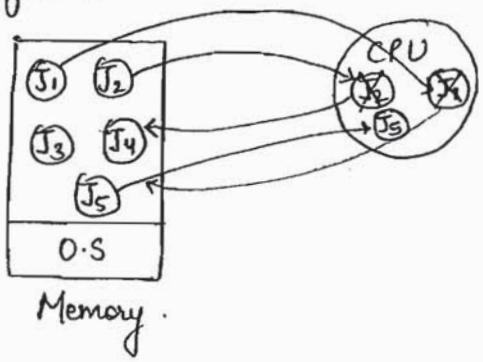
### (2). Multiprogramming O.S.:




- If the job is leaving the CPU to perform IO operation, then another job which is ready for execution will be scheduled onto CPU.
- Advantage
  - Increased CPU Utilization.
  - Increased throughput of the system.

**Exp**: Windows, UNIX.

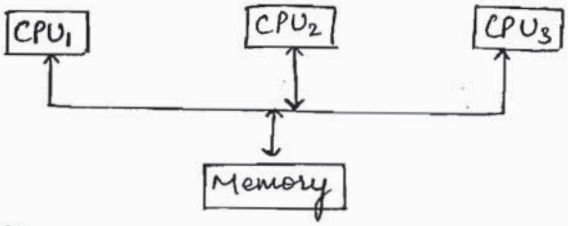
## (3). Multitasking OS :




- Multitasking is an extension of multiprogramming OS.
- The jobs will be executed in the time sharing mode.

Exp: Windows, Unix

## (4). Multiprocessor Systems :

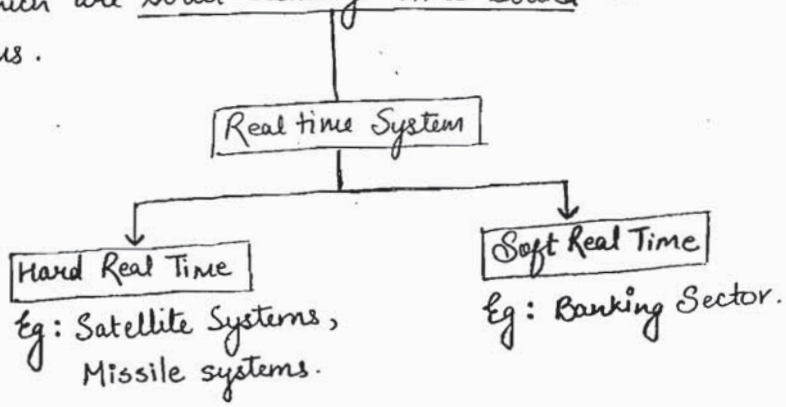



- Advantage
  - Increase the throughput of the system
  - Reliability
    - → Fault Tolerant Systems.
  - Economical.

Exp: UNIX.

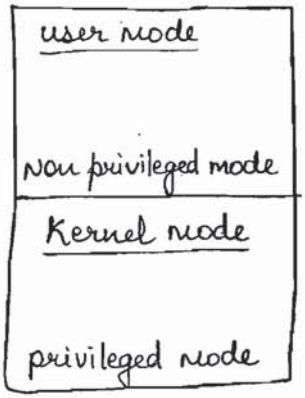
## (5). Real Time Systems :

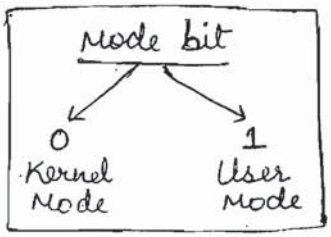
- The systems which are strict deadly time bound are called as real time systems.




Exp: Sx works, Vx works, RTO's.

# Dual Mode Operation:




- In the hardware Level, the instructions are executed by using dual mode operation like
  1. user mode / Non privileged mode
  2. Kernel mode / priviledged mode / system mode / monitor mode.
- The dual mode operation is used in order to provide protection & security to the user programs. and also to the operating system from "errant users" (unauthorized users).
- It is purely the decision of the operating system in which particular mode, the instruction has to be executed.
- The mode bit is used to identify in which particular mode, the current instruction is executing.
- The privilised instructions are executed in the kernel mode & Non privilised instructions are executed in the user mode.
- In the Boot time, the system always starts only in the Kernel mode.
- The operating system always runs only in the kernel mode

**Note:** The mode switching takes very less time compared to process switching.

## Privileged Instructions :

1. I/O operation
2. Context Switching
3. Disabling the interrupts
4. Set the time of clock.
5. Clearing the memory map.
6. Changing the memory map.

## Non-Privileged Instructions :

1. Reading the time of the clock
2. Reading the status of the processor.

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

## Array :

(EX1) int a[10] = {10, 20, 30, 40, 50 ..... 100}

(declaration)

↓

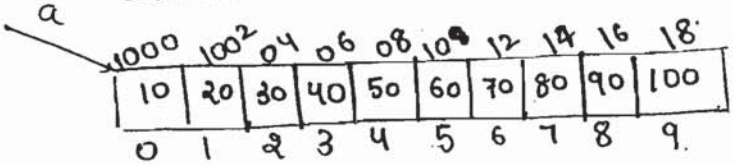
identifier / variable / fn name.

iden

→ scan from left due to LL or LR parser
(Top down, Bottom up)

int a [10] — a is array of (size 10) having elements as integer.

winner




a is array which contain 10 elements where everyone is integer

integer = 2 Bytes

j - i → gives elements before j

j - i + 1 → gives elements including j

X { (int a) [10] — stop.
a is a integer

(Bracket) ( ) have Left to right associative.
(If come first it will be done).

print a : 1000 will print (array name print gives Base address will print)

a — array name.

print variable name : print the value of that value

$$\boxed{LOC(a[3]) = 1000 + (5-0) * 2 = 1010}$$

for LOC → (&) reference.
* Deference

$$\boxed{LOC(a[9]) = 1000 + (9-0) * 2 = 1018}$$

using this formula anyone can be accessed

⇓

Random access.

EX2 A[75 ..... 330] 330 − 75 + 1 = 276 elements

starting index → 75 ; ending index → 330

BA = 1000 , c = 10 (size)

$[a[290]] = 1000 + 290 \times 10$ ..... (290 − 75)

$1000 + 2900$ ..... 215 × 10

$(a[290]) = 3900$ (3150) ..... 2150 + 1000 = 3150

EX3 A[−90 ... 500] (591) 500 − (−90) + 1 ⇒ 591

BA = 0 , c = 5 Bytes

$LOC(A[393]) = 0 + (92 - 393) * 5$

$= 0 + (393 + 90) * 5$

$= 2415$

393 − 92 = 301 ; 301 / 4 = 485 , remainder 1 ; 485 * 5 = 2425

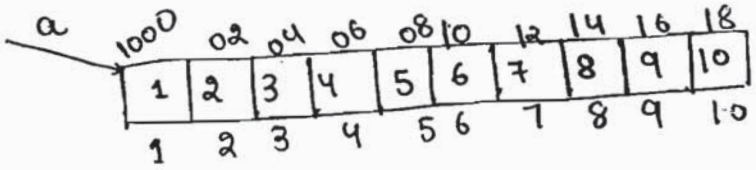
NOTE

A[lb .... ub]

BA , c.

$LOC[A[i]] = BA + (i - lb) * c$

$LOC(a[3]) = 1000 + (5-1) * 2 = 1008$ ((5−1) is the extra)




Note.

By default array index start from 0 not from 1 because no need to calculate offset value (substraction)

(No need to perform extra substraction)

## 2D-Array:

$4-1+1 = 4$ rows

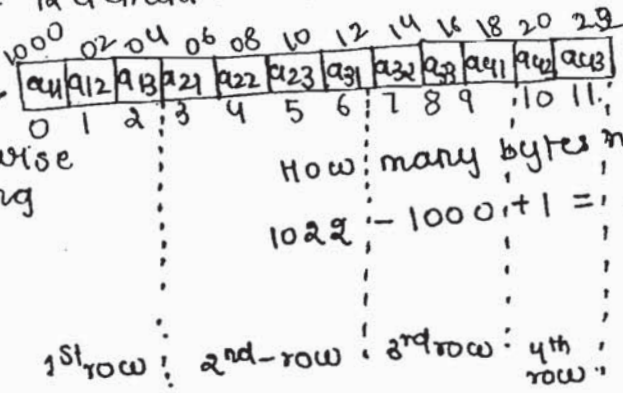
int A[1 ..... 4, 1 ..... 3]

$3-1+1 = 3$ columns.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | a11 | a12 | a13 |
| 2 | a21 | a22 | a23 |
| 3 | a31 | a32 | a33 |
| 4 | a41 | a42 | a43 |

Row major order

$4 \times 3 = 12$ elements = 12 slots needed to store




How many bytes needed

$1022 - 1000 + 1 = 22 + 1 = 23$ Bytes.

$LOC(A[4][3]) = 1000 + 3*4(2)$

$= 1000 + 12*2$

$= 1024$

$(4-1)(3-1)$

$3*2*2$

$6*2$

$= 12$

$LOC[A[4][3]] = 1000 + [(4-1)*3 + (3-1)]*2$

$= 1022$

$(3-1)*4(2)$

$2*8$

Ans $LOC(A[2][3]) = 1000 + [(2-1)*3 + (3-1)]*2$

$= 1000 + [3 + 2]*2$

$1000 + 10$

$= 1010$

r

$(3)*3$

$9 + 2$

$11*2$

$1000 + 22$

$= 1022$

ex(2) $A = [29 \cdots 76, 93 \cdots 200]$.

row $\Rightarrow 76-29+1 = 48$ $\quad \Rightarrow 108$ c.

$(j-i)*j +$

BA = 1000, c = 10 Row major order

$LOC(A[70][190]) = 1000 + [(70-29)*108 + (190-93)]*10$

$= 1000 + [4428 + 20]*10$

$= 1000 + 44480$

$= 45480.$

| | | | |
|---|---|---|---|
| 76 | 200 | 70 | 108 |
| 29 | 93 | 29 | 41 |
| 47 | 107 | 41 | 108 |
| | | | 432 x |
| | | | 4428 |

| | |
|---|---|
| 44480 | |
| 1000 | |
| 454 | |

40 250

ex(3) $A[-200 \cdots +200 , -300 \cdots -150]$

$200-(-200)+1$ $\quad -150-(-300)+1$

$-150+300+1$

$150+1$

$=151$

$BA = 0 \; , C=1 \; , RMO$

$$LOC[A[-3][-170]] = D + [(-3-(-200))*151 + [-170-(-300)]]$$

$$= 197*151 + 130$$

$$= 29747 + 130$$

$$= 29877$$

200 − 3 = 197

300 − 5 = 295

300 − 170 = 130

197 × 151 = 29747

NOTE: $ub_1 - lb_1 + 1 = n_r$

$A[lb_1 \cdots ub_1 , lb_2 \cdots ub_2]$ $\quad n_c$

$BA, C, RMO$

$$LOC(A[i][j]) = BA + [(i-lb_1)*n_c + (j-lb_2)]*C$$

NOTE:

column Major order $= 48$ $= 108$

ex(4) $A[29 \cdots 76 , 93 \cdots 200]$

$BA = 1000 \; , C = 10 \quad CMO$

$$L(A[70][190]) = 1000 + [(190-93)*48 + (70-29)]*10$$

$$= 1000 + [87*48 + 41]*10$$

$$= 47970$$

190 − 93 = 87

70 − 29 = 41

87 × 48 = 41

last element address in RMO = last elem add in CMO

$A = [-200 \cdots +200 \;(\Rightarrow 401),\; -300 \cdots -150 \;(\Rightarrow 151)]$

BA = 0  C = 1  RMO

$LOC(A[-3][-170]) = 0 + [(-170+300)*401 + (-3+200)]*1$

$= 130 \times 401 + 197$

$= 52327$

300 − 170 = 130

401 − 13 = 388

NOTE: $A(lb_1 \cdots ub_1 \; lb_2 \cdots ub_2)$  BA, C, RMO, $nr = n_2$

$$LOC(A[i][j]) = BA + [(j - lb_1) * nr + (i - lb_1)]C$$

1203 × 401 = 52130 + 197 = 52327

3D-array

Ex(1)

$A[23 \cdots 49,\; 2 \cdots 19,\; 11 \cdots 29]$

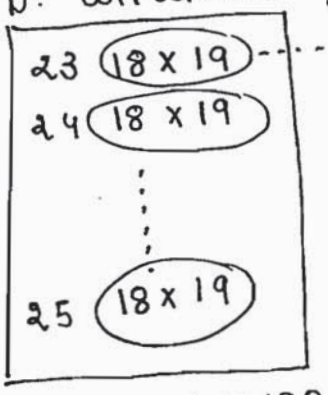
no of 2D = $49-23+1 = 26+1 = 27$

$n_1 = 19-2+1 = 17+1 = 18$

$n_2 = 29-11+1 = 18+1 = 19$

$18 \times 19 \Rightarrow$ 2D.

3D: collections of 2D.




- collects of elements ⇓ 1D
- collection of 1D ⇓ 2D

BA = 1000, C = 10, RMO

size of this 18 x 19.

$LOC(A[40][15][20]) = 1000 + [(40-23)*27 + (15-2)*19 + (20-11)]*10$

40 − 23 = 17

27 − 9

$= 61700$

column major order

$$LOC(A[40][15][20]) = 1000 + [(40-23)*18*19 + (20-11)*18 + (15-2)]*10$$

$$= 60890.$$

NOTE:

$A[lb_1 \cdots ub_1, lb_2 \cdots ub_2, lb_3 \cdots ub_3]$

BA, C, RMO

$$LOC(A[i][j][k]) = BA + [(i-lb_1)*n_c*n_r + (j-lb_2)*n_c + (k-lb_3)]C$$

For ROM

CMO $LOC(A[i][j][k]) = BA + [(i-lb_1)*n_c*n_r + (k-lb_3)*n_r + (j-lb_2)]*C.$

loops take time
this is formula
will not take any time.

Lower triangular Matrix

$A[1\ldots4, 1\ldots4]$

Triangular Matrix ⇒ should be square matrix.

$j > i$ element value 0

$j \le i$ Non zero value

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | $a_{11}$ | 0 | 0 | 0 |
| 2 | $a_{21}$ | $a_{22}$ | 0 | 0 |
| 3 | $a_{31}$ | $a_{32}$ | $a_{33}$ | 0 |
| 4 | $a_{41}$ | $a_{42}$ | $a_{43}$ | $a_{44}$ |

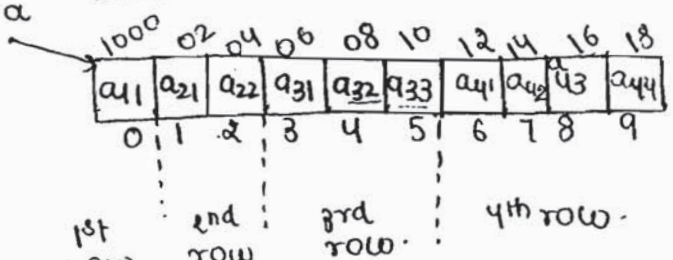
Total Non zero $\Rightarrow 1+2+3+\cdots\cdots n$

$$\frac{n(n+1)}{2}$$

Total slots $= \frac{n(n+1)}{2}$ for $n = 4$

$= 4*5/2 = 10$ slots

## Row major order




$$LOC(A[4][3]) = 1000 + \frac{(i+j)-1}{(3+3-1)*2}$$

$$= 1000 + \left[\frac{(4-1)(4-1+1)}{2} + (3-1)\right] * 200 + 5*2$$

$$= 1000 + 10$$

$$= \underline{1016}.$$

$$\frac{3*4^2}{2} = \underline{1010}$$

$$1000 + \underbrace{(6+2)}_{8*2} = 1000 + (4+3) - 1$$

$$= 1000 + 12$$

$$6*2$$

ex② $A[39 \cdots 80, 39 \cdots 88]$ $\Rightarrow nr = 50$, $\Rightarrow nc = 50$.

should be same.

88
39
49+1
50

$BA = 1000$, $c = 10$, LTM, RMO.

$$LOC[A[72][55]) = 1000 + \left[\frac{(72-39)(72-39+1)}{2} + (55-39)\right] 10$$

$$= 6770$$

72-
39
33

17
*34
2

2 | 33
51
51 x
561
16
5770
1000
6770

55
39
16

ex③ $A[\underbrace{50}_{add=200} \cdots 150, 250 \cdots 350]$ $nr = 101$, $nc = 101$

200 3

$BA = 1000$
$c = 10$
RMO, LTM.

250 ···· 350
50 [50,251 ]
51
151 [ ]

X

$LOC(A[100][275])$ or substract 200 Here.

$LOC(A[300][275]) = 1000 + [(300-200) + (275$

X

$= 1000 + [1010$

$= 1000 + 101$

$= \underline{102250}$.

$i-1$
$6-2$
$= 4$

$= 6-2$
$= 4*2$
$=$

$241$
$4+1$
$5*2$.

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

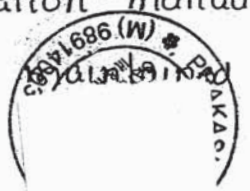
Software is a program, when it is executed the associated functionally must be satisfied.

S/w is a data structure used to manipulate the information

S/w consist the operational procedure, used to indicate the operational guideline of the s/w....

ie... user manual, beginners guide, system overview... etc

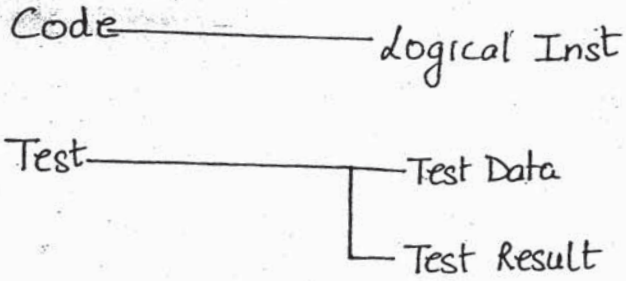
Software consists the documentation manual, used to indicate the list of the activities maintained to develop the s/w....



Software Consists the documentation manual, used to indicate the list of the activities maintained to develop the s/w

ie....

- Analysis Specification
  - Formal specification
  - Context Program
  - Data flow Diagram
- Design
  - Flow chart
  - ER Diagram




S/W is a combination of Program, operational Procedure & Instruction Manuals, Finally we can conclude the S/W is a logical Component rather then Physical Component.

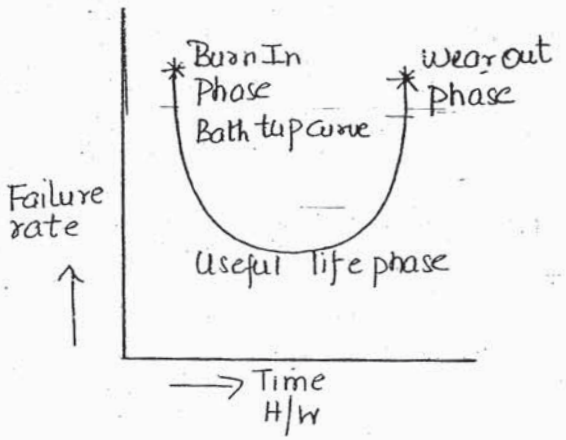
Software = Programs + operational + Documentation, Procedure Manuals

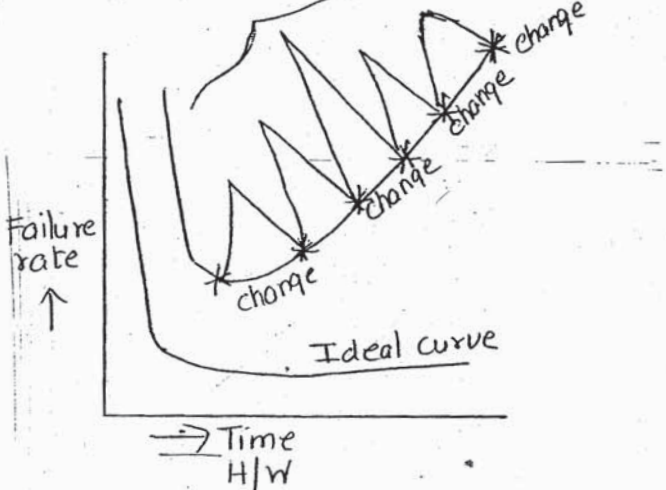
## Character of Software-

1. S/W. is developed or engineered but not manufactured in the classic sense.
   i) S/W development doesn't have the assembly line.
   ii) Logic of the program is developed only once & that can be copied into any no. of the copies.
   iii) Physical Components consists the assembly line to manufacture the product, during this process every time raw materials are included.
   iv). After implementation of design document if it is generate the logical component called as the development

Process else that process is called as manufacture.

2. S/W doesn't WEAROUT but DETERIORATE → Deterioration




The life cycle of H/W component consist of the 3 stages :-

## Burn in Phase

In the early stage of development due to the more numbers of errors the product failure rate is high.

In the Burn-In- Phase the product is present in the developer's site.

After various test operations the failure rate is decrease & it will be established at one point.

Later deployee the product into customers place.

## Useful life Phase

In this Phase the product is there in operational state.

## Wear Out Phase

After the continuous users of the product over a period of a time the life time of the product is decreases due to the environmental changes.

ie... Temp, Vibrations, Dust Etc...

Therefore the failure rate of the product increases.

NOTE :-

When the H/W component is undergoes wear out condition, then replace the component with new component.

S/W is undergoes deterioration

ie... At the early stage of operation the product failure rate is high.

After the testing operation the product will be deployed in to the customers site.

S/W doesn't affected by the environmental changes but the customer requirements are not static. When the customer requirements are changed frequently then the maintenance cost of the S/W becomes twice or thrice than the development cost. This condition is called deterioration.


Note :-
When the S/W is deteriorate then Re-Engineer the S/W

3. Industry is moving towards component based development. Still the S/W is customer build.

Component is a reusable code or error free code or risk free code or fully tested code.

During the Physical component manufacturing process we can directly use the component without changes.
ie .... Working models contains different I.C to implement the model directly purchase the I.C from market.

In the S/W development process, supplication to application the functionality will be differ therefore the S/W components are undergoes customization according to project functionally.

In the S/W development 4 Types of Components are used :-

1. Off the shelf

These components are extracted from 3rd party library

2. Fully - Experienced

3. Partial Experienced

~~These components are extracted~~ from the own org library.

4. New Component

New Component functionality is developed from the base line.

## S/W Applications:-

There is different applications domain used as the S/W to satisfy the various high level objectives.

1) System Software: is a program used to manage the other program or provide the services to other program ie: OS, Compilers, editors, drivers... ETC

2) Real time S/W: This s/w is used to monitor, control and analyze the various real time events when they are occur. Eg: Whether forecasting uses the real time s/w to control the temp, humidity & other environmental parameters to forecast the weather.

3) Embeded S/W: The s/w is placed into the permanent memory (RAM) of the product & it is used to control the various function of product.

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# THEORY OF COMPUTATION

- GODEL : Logic is limited
- Turing : Model for computation
- POST
- Chomsky Heirarchy

→ 2 types of computer

Acceptors: Yes/No, given lang. Accept or Not Accept.

Transducer: computational
x is given f(x) can be computed.

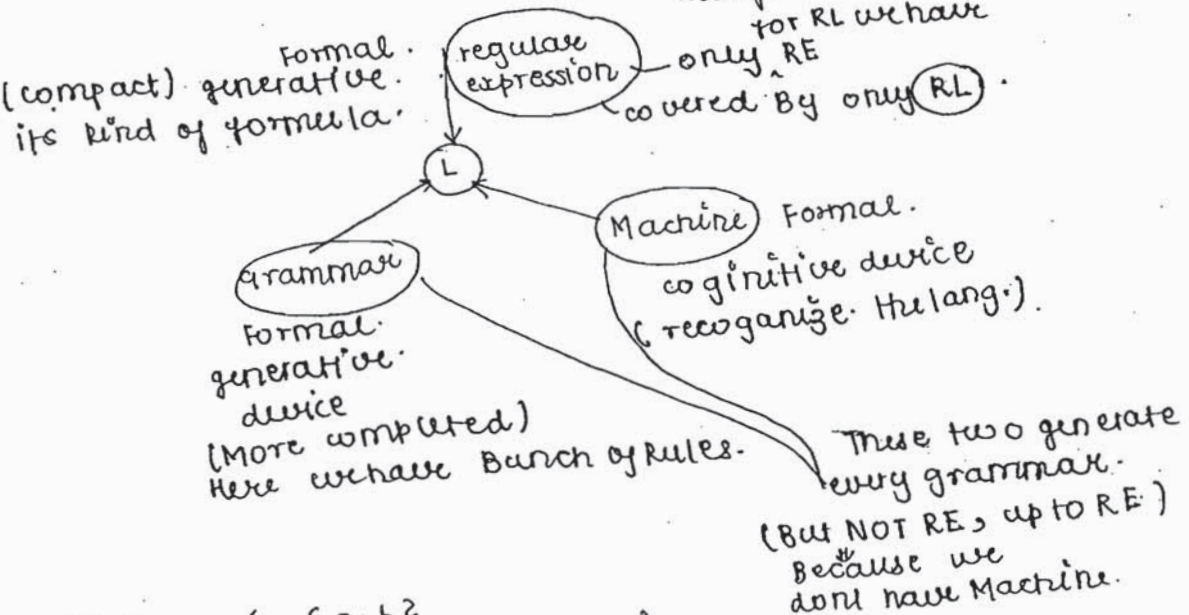
- every problem has associated with language.. we bother about acceptance of language. if we can accept the language. we can say problem is solvable.

## chapter-0

- Language : collection of strings
  - grammar:
  - Machine:

Language can be described by grammar.

- (L) Informal (can't list all the things in language)




1. Alphabet = $\Sigma = \{a, b\}$
2. String
3. concatenation
4. Reversal
5. Length of a string
6. NULL string. = "$\epsilon$"
7. PREFIX
8. SUFFIX
9. substring
10. substring.
11. Powers of a string $(\omega)^n$.
12. $\Sigma^*$, $\Sigma^+$
13. $L \subseteq \Sigma^*$.
14. CHOMSKEY HEIRARCHY
14. ~~operat~~ Representations of Language

- Language

1. alphabet:

14. Representation of Language 6 → 3 Formal, 3 Informal.

15. operations on language.
union, intersection, $\bar{L}$, $L_1 - L_2$, $L_1 \oplus L_2$

16. concatenation of Lang.

$$L_1 \cdot L_2 = \{uv \mid u \in L_1, \& v \in L_2\}$$

17. $L^R = \{w^R \mid w \in L\}$ Reversal of language.

18. $L^*$, $L^+$

$L^*$ contain $\epsilon$ every possible combination of strings.

$L^+$ Not contain $\epsilon$

$L^+ = L^* - (\epsilon)$ ⟶ This is Not correct statement.

• **Alphabet**: a Non-empty finite set of symbols.

$\Sigma = \{\}$ Not alphabet

- $\Sigma = \{a\}$, $\Sigma = \{1\}$, $\Sigma = \{2\}$ $\Sigma = \{3\}$ (1 symbol) unitary alphabet
- $\Sigma = \{a, b\}$, $\Sigma = \{1, 0\}$ Binary alphabet (2 symbol).

$\Sigma = \{1, 11, 111 \ldots\}$ This is Not allowed, No of symbols should be finite.

$\{0, 1, 2\} = \{0, 1, 2\}$

both alphabet same, order dont matter.

symbol $\Sigma = \{\underbrace{01}_{\text{compound symbol}}, 10\}$.

$\Sigma = \{01, 10, 1, 0\}$ This is Not valid symbol 01 or 10 cant break further.

• **String**: sequence of 0 or more finite symbols taken from the alphabet

sequence : order is important.

$\Sigma = \{a, b\}$.

$a^{100}$ is valid string ? ⇒ Yes.

aaa ... 100 times valid.

aabb(1) valid ? ⇒ No symbols can be taken from alphabet.

baab = baba Not equal string, 'sequence should be follow'

$(ab)^2 \neq a^2b^2$ Not valid in TOC.

abab ≠ aabb

• **concatenation**:

if u = 01

v = 100

uv = 01|100 concatenation

Here $uv \neq vu$.

for all (u, v)

where u = 00

v = 000.

$uv = vu$ True Here

so Not for all (u, v); $uv \neq vu$.

- it is associative
  $u(vw) = (uv)w$
- Not communicative.

- The length of u·v will always be equal to (u+v).

u = 01 = 2

v = 100 = 3

uv = 01100 ⇒ length is (2+3) = 5

- The length of v i

v = 100 $v^R = 001 \Rightarrow v^R \neq v$

For all u and v $uv \neq vu$ False

• $v = v^R$ iff $v$ is a palindrome

Palindrome

- even (because length is integer). : EP
- odd. : OP.

Palindrome Language $= \{ \underset{EP}{ww^R} \cup \underset{OP}{wxw^R} \mid w \in \Sigma^*, x \in \Sigma \}$ ($x$ is 1 bit)

$\Downarrow$ $\Downarrow$

1001 10001

$\{ w \in \Sigma^* \mid w = w^{\overline{x}} \}$

• Properties of Reversal

$(v^R)^R = v$

(Reversal of Reversal).

$(u \cdot v)^R = v^R u^R$

$(xyz)^R = z^R y^R x^R$.

• Length :

No of symbols present in strings.

if $\Sigma = \{0,1,2\}$.

→ How many 5 length string possible. if $\Sigma = \{a,b\}$.

0 length → 1 (i.e $\varepsilon$)

1 length → a, b → $|\Sigma| = 2$

2 length → ab, ba, aa, bb → $|\Sigma|^2$. (cardinality of $\Sigma$.)

⋮

n length → $|\Sigma|^n$.

So of length '3' $2^3$ ⇒ for $\Sigma = \{a,b,c\} \Rightarrow |\Sigma| = 3$.

length '5' $3^5$

$ww^R$ How many even palindrome of length 10.

$\Sigma = \{0,1\}$

$\overline{2}\ \overline{2}\ \overline{2}\ \overline{2}\ \overline{2} \mid - - - - -$

$(2^5)$.

even palindrome $= |\Sigma|^{n/2}$

- ODD ⇒ length 13. check even → 12. $\Sigma = \{0,1\}$

$|\Sigma|^{\frac{12}{2}}$ $\quad |2|^6 \Rightarrow 2^6 \times 2$ odd palindrome.

upto length 10, even palindrome. $\Sigma \{0,1\}$

| | | |
|---|---|---|
| 0 | length ⟶ | $2^{0/2} = 1$ |
| 2 | " | $2^{2/2} = 2$ |
| 4 | " | $2^{4/2} = 4$ |
| 8 | " | $2^{8/2} = 2^4$ |
| 10 | " | $2^{10/2}$ |

- **NULL string:**
  it is only string of length '0'

lang with Null string. $|\{\varepsilon\}|$ ⟶ cardinality.

$|\varepsilon| = 0$.

$|\{\}| = 0$.

empty language.

- language has cardinality.
  where
  Null Reverse is Reverse
  Null is identity element for concatenation.

- $(\Sigma^*, \cdot, \varnothing)$ ← groupoid, operator, semi-group.
  ↓ Algebra.

- $|\{0,1\}|^5 \longrightarrow |\{0,1\}|$
  How many fn possible
  all string of length 5.

$|\Sigma|^5 \longrightarrow \Sigma$ possible function.

cardina. $= 2^5$ $\qquad 2$.

$|B|^{|A|}$ function possible $= \underline{\underline{2^{(2^5)}}}$

7. PREFIX : Prefix is set of strings. so P(w) is

$P(w) = \{u \mid w = uv\}$

- prefix of 001 = $\boxed{\epsilon} \cdot 001$
  = $\boxed{0} \cdot 01$
  = $\boxed{00} \cdot 1$
  = $\boxed{001} \cdot \epsilon$
- Null string is prefix of every string
- and $\underline{w}$ is also the prefix of $\underline{w}$ string

8. SUFFIX

$S(w) = \{v \mid w = uv\}$

$\epsilon \cdot \boxed{0\,0\,1}$

1, 01, 001, $\epsilon$.

string w of length ⇒ $|w| = n$ so $(n+1)$ prefix are there.

- $|\text{Prefix}(w)| = n+1$.
- $|w| = n$
- same for suffix
- for any length prefix or suffix is unique.

for $\Sigma = \{a, b, c, d, e\}$

How many prefix of length 3 = 1
always 1 upto the length of string.

- 000 - $\{\epsilon, 0, 00, 00\}$.

$\boxed{\text{prefix}(w) \cap \text{suffix}(w) \supseteq \{\epsilon, w\}}$ true. (⊇ : superset)

'⊂' wrong
'=' wrong

- substring : $\{x \mid w = uxv\}$ — By making $u = \epsilon \rightarrow$ prefix, $v = \epsilon \rightarrow$ suffix, $v, u = \epsilon \rightarrow$ allowing middle strings

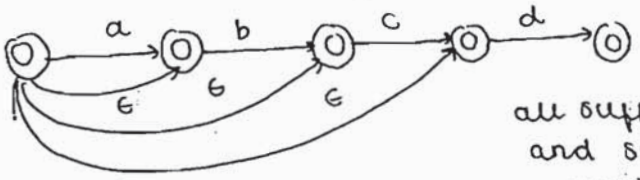
  any sequence of 0 or more consequtive symbols in 'w'

  for abcde

  $\epsilon \rightarrow$ ✓
  ab → ✓
  bcd → ✓
  cde → ✓
  bce → x sequence failed.

- substring includes all prefix and suffix.

$\boxed{\text{prefix}(w) \cup \text{suffix}(w) \subseteq \text{substring}}$




all suffix prefix accept and substring also accepted (n+1) states needed.

10. $w^n$ : powers of string

Here it is concatenation.

$w^0 = \epsilon$ → for concatenation $\epsilon$ is identity for $w^0 = \epsilon$

$w = 001$

$w^1 = w$

$w^2 = w \cdot w$

$w^3 = w \cdot w^2$.

$w^{-1}$ → has no meaning

because $u \cdot u^{-1} \neq \epsilon$

There is nothing which adds in string, so strings becomes, string length always ↑.

→ $(\Sigma^*, \cdot, \emptyset)$ → Monoid

closure → groupoid. ✓

→ TOC dont allow -ve Numbers.

(powers). $a^m$ where $m > 0$.

- powers of w are commutative.

$(001)^0 = \epsilon$

$(001)^1 = 001$.

$(001)^2 = 001001$.

$(001)^3 = \underline{001}\,\underline{001}\,\underline{001} = (001)^2 \cdot (001)$

$(ab)^{18}\,(ab)^2 = (ab)^{20}$

$(ab)^2\,(ba)^2 \neq (ab)^4$

$a^m a^n b^n c^n = a^{m+n} b^n c^n$.

11 $\Sigma^*$ : $\quad \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$

$\Sigma^0 = \epsilon$

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$.

$\boxed{\Sigma^+ = \Sigma^* - \epsilon}$ positive closure of '$\Sigma$'

'*' closure (Kleene closure) $\Rightarrow \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$

Kleene star operator.

- $\Sigma = \{a\}$

$\Sigma^* = \{\epsilon, a, aa, aaa \ldots\}$.

- Language is infinite but alphabet is finite

# ENGLISH

Sapna

Ludhani.Sapna@gmail.com
9911629948

gate
- Critical Reasoning
  - Inferential Reasoning
  - Logical deduction
- Parajumbles
- Reading Comprehension

gate
- Vocabulary
  - Analogies
  - Sentence Completion
  - One word Substitutions
  - Synonyms
  - Antonyms

gate
- Grammar
  - Spotting the Error
  - Sentence Improvement

## Analogy {Relationship}

Q. Gladiator (warrior) : Arena (Plat form)

a) Dance : Stage
b) Commuter : train
c) Teacher : classroom
d) ✓ Lawyer : courtroom

Gladiator
a) performer
b) fighter
c) opponent
d) End result → win / loose.
(Traveller - Commutator)

Q. frequency [Antonym ?]

a) Periodicity
~~b)~~ ✓ Rarity
c) Gradualness
d) presistency (continuity) Perseverance
↳ Continuous determinat°

Q. Children : Pediatrician (child specialist)

a) Adult : Orthopaedist (Borns specialist)
~~b)~~ ✓ females : Gynacologist (female specialist)
c) Kidney : Nephrologist (deal with kidney & Urine)
d) Skin : Dermatologist (deal with Skin)

Q. Nocturnal : Bat.
(active at night)

~~a)~~ ✓ Amphibian : frog → Stay land or water
~~b)~~ d Sly : cat → cunning (deceitful)
~~c)~~ d Carnivorous : cow → eating flesh)
d) Aquatic : Liz. ↳ Lizard.
↓ (live in water).

5. Xenophobia : foreigners. (~~least~~ similar)
Fear of foreigneers & strangers

a) Bibliophobia : Books
b) Anglophobia : English
c) Hemophobia : Blood
d) Claustrophobia : Hight (fear of height is Acrophobia)
↳ (Fear of Confined or Closed Spaces) or conjusted place.

{Phobia → fear
{Mania → Obession, (craze)

Vibrant (full of happyness, Energy)

~~Cheerful~~
~~depressed~~

Clumsy → ∘ dirty, Unpleasant
∘ unskilled

| Synonyms | Antonyms |
|---|---|
| ∘ Lively | ∘ Lazy |
| ∘ Vivacious | ∘ Lethargic |
| ∘ Vigourous | ∘ Sluggish. ↳ slow & Inactive |
| ∘ Enthusiastic | |
| ∘ Energetic | ∘ Dormant. |
| ∘ Passionate | ∘ Indolement. |
| ∘ Zealous. | ∘ Dizzy |
| ∘ Zestful | ∘ Drowsy ↳ Sleepy |
| ∘ Exuberant. | |
| ∘ Active. | ∘ Lackadaisical |

Fatique
↳ tiredness.

Liberal => open minded.

## Philanthropist (A lover of mankind).

- Phil : love.
- Anthrop : Mankind.

### Synonyms.

- Kind
- Generous
- Altruist
  - ↳ selfless
- Benevolent → Kind & Charitable.
- Benign → kind & good natured.
- Magnificient
  - ↳ kind / huge structure
- Magnanimous
  - ↳ Large hearted.

Liberal → open minded. Hypocrite → One who does not follow what he preaches. (follow double std).

### Conservative.

- Orthodox.
- Parochial
- Narrow minded
- Dogmatic
- Stickler
- Adamant.

## Extrovert

- Social
- Sociable
- Gregarious
  (Like to live in group)
  → Flock

## Introvert

- Reserved
- Reclusive
- Shynatured
- Asocial
- Aloof (Alone)
- Segregated
- Loner (Lonely or Alone).

- Ambigous (Indicided State)
  Ambi : both
  → & Dilemma
  Confusion between two choice

* Ambivert => Both Introvert & Extrovert.

Amiable => Lovable, Friendly, Pleasant in Nature

Amity → Love,

*Fastidious => Perfectionist, Extremely careful about detail difficult to please

Garrulous => Talkative.

Bibliophile ⟶ Lover of Books.

Ambidexterous ⟶ Use Both hands with equal skill.

Q. Nadir [Antonyms] 2. Medicine : Health (similar pair)
↳ Zenith

a) ~~Highest~~
b) Lowest
c) Medium
d) Integrath

a) Science : Experiment
b) Money : Happiness
c) Wealth : Peace
d) Educath : Knowledge.

Zenith → Highest point, Apex, Summit, Peak, Pinnacle, Acme, Apothesis.
↓
opp
↓
Nadir

# Hindbookcenter

GET IT ON Google Play

**ALL NOTES BOOKS AVAILABLEALL STUDY MATERIAL AVAILABLE**

**COURIERS SERVICE AVAILABLE**

## MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

**ESE , GATE, PSUs BEST QUALITY TOPPER HAND WRITTEN NOTES**

**MINIMUM PRICE AVAILABLE @ OUR WEBSITE**

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

## IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ **NOTE;- ALL ENGINEERING BRANCHS**

➢ **ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE**

## PUBLICATIONS BOOKS -

**MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK**

**RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON…OTHERS**

**HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE**

| | | | |
|---|---|---|---|
| Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 | Shop No: 46<br>100 Futa M.G. Rd<br>Near Made Easy<br>Ghitorni, New Delhi-30 | F518<br>Near Kali Maa Mandir<br>Lado Sarai<br>New Delhi-110030 | Shop No.7/8<br>Saidulajab Market<br>Neb Sarai More,<br>Saket, New Delhi-30 |

**Website: www.hindbookcenter.com**

**Contact Us: 9711475393**

# NUMBER SYSTEM

## Factors:-

Factors are the set of numbers which is divide a given no. completely.

Ex:-1 $72 = 2^{3} \times 3^{2} = 4 \times 3 = 12$

(3+1) (2+1)

1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72.

Ex:-2

$$120 = 2^{3} \times 3^{1} \times 5^{1} = 4 \times 2 \times 2 = 16$$

1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120

## Total Factors / Divisors :-

$$N = a^{p} \times b^{q} \times c^{r} \ldots\ldots$$

$$Tf = (p+1)(q+1)(r+1) \ldots\ldots$$

where, a, b, c .. are distinct prime numbers and p, q, r ar natural numbers.

Ex:-3 9000

find the Tf, odd factor, even factor?

$$N = 2^{3} \times 3^{2} \times 5^{3}$$

i) $Tf = 4 \times 3 \times 4 = 48$

ii) odd f :→

To Remove the 2 from the System and use same procedure.

Odd factor = $2^3 \times [3^2 \times 5^3]$

= $3 \times 4 = 12$

Even factor = $48 - 12 = 36$

Q1. How many factors of number 72 are multiple of 6.

$72 = 2^3 \times 3^2$

$= \underbrace{2^1 \times 3^1}_{6} \left[\underbrace{2^2 \times 3^1}_{12}\right]$

$= 6 [1, 2, 3, 4, 6, 12]$

Tf = $3 \times 2 = 6$

2 | 72

Q2. How many factors of number 120 are multiple of 12.

$120 = 2^3 \times 3^1 \times 5^1$

$= 2^3 \times 3^1 \left[\underbrace{2^1 \times 5^1}_{10}\right]$

$= 12 [1, 2, 5, 10]$

Tf = $2 \times 2 = 4$

Q3. How many factors of number 9000 are multiple of 30.

$9000 = 2^1 \times 3^1 \times 5^1 [2^2 \times 3^1 \times 5^2]$

Tf = $3 \times 2 \times 3 = 18$

Prime and composite factors:-

$72 = 2^3 \times 3^2$

$T_f = 4 \times 3 = 12$

$P_f \longrightarrow 2$ for [2, 3]

$C_f \longrightarrow 9$

$NPNC \longrightarrow 1$

Total = 12

Ex:-2 $120$

$2^3 \times 3^1 \times 5^1$

$= 4 \times 2 \times 2 = 16$

$P_f \longrightarrow 3$ for [2, 3, 5]

$C_f \longrightarrow 12$

$NPNC \longrightarrow 1$

Total → 16

$$T_f = P_f + C_f + 1$$

*** Sum of factors

$N = a^p \times b^q \ldots\ldots$

$$\text{Sum of factors} = \left(\frac{a^{p+1}-1}{a-1}\right)\left(\frac{b^{q+1}-1}{b-1}\right)\ldots\ldots$$

$N = 72 = 2^3 \times 3^2$

Sum of factor $= \left(\frac{2^4-1}{2-1}\right)\left(\frac{3^3-1}{(3-1)}\right)$

$= 15 \times 13$

$= 195$

## Product of factor

$$P.F = [N]^{\frac{Tf}{2}}$$

Ex:- 72

$Tf = 2^{3} \times 3^{2}$

$= 4 \times 3^2 = 12$

So, $PF = (72)^{\frac{12}{2}}$

$= (72)^6$

Ex:-2 $N = 36$

$= 2^2 \times 3^2 = 3 \times 3 = 9$

So, $PF = (36)^{9/2}$

$= (36)^{4.5}$

## Factorials

$1! + 2! + 3! + \cdots\cdots + 99!$ then what would be the unit place/Digit/last digit/last place of the number (u)?

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

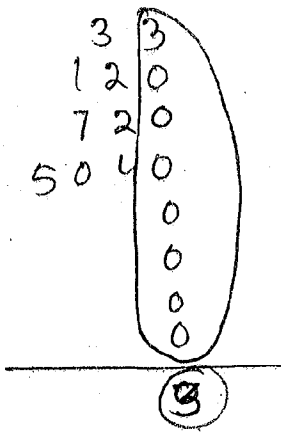
$6! = 6 \times 5! = 720$

$7! = 7 \times 6 \times 5! = 5040$

$1! + 2! + 3! + 4! + (5! + 6! + 7! + 8! \cdots\cdots + 99!)$

$1 + 2 + 6 + 24 \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0$

$= 33$




Ex:-2 100! factorial contains howmany zero.

$100! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times \cdots \times 100$

$\frac{100}{5} = 20$ [5, 10, 15, 20, ..... 100] $\simeq 5^1$.

$\frac{20}{5} = 4$ [25, 50, 75, 100] $\simeq 5^2$

Total = 20 + 4 = 24 → no. of zero.

Ex:-3 OR

How many power of 3.

$100! = 1 \times 2 \times \overset{\sim}{3} \times 4 \times 5 \times \overset{\sim}{6} \times 7 \times 8 \times \overset{\sim}{9} \times 10 \times \cdots \times 99 \times 100$

$\frac{100}{3} = 33$ [3, 6, 9, 12, ..... 99] $\simeq 3^1$

$\frac{33}{3} = 11$ [9, 18, 27, ..... 99] $\simeq 3^2$

$\frac{11}{3} = 3$ [27, 54, 81] $\simeq 3^3$

$\frac{3}{3} = 1$ [81] $\simeq 3^4$

48

OR $\frac{100}{3} + \frac{33}{3} + \frac{11}{3} + \frac{3}{3} = 33 + 11 + 3 + 1 = 48$

Ex:- $\frac{100}{7} = 14$

$\frac{14}{7} = 2$

So $14+2 = 16$ nos.

Ex:- $\frac{100}{15} = 6$ & $\frac{100!}{(15)^n}$

$= [15, 30, 45, 60, 75, 90]$

Bullshit

$$\frac{100!}{(10)^n} = \frac{100!}{(2\times5)^n}$$

$$100! = 1\times2\times3\times4\times5$$

$$\frac{100!}{(15)^n} = \frac{100!}{(3\times5)^n}$$

$$100! = 3^{48}\times5^{24}$$

$$= 3^{24}\times3^{24}\times5^{24}$$

$$= (3\times5)^{24}\times3^{24}$$

$$= (15)^{24}\times3^{24}$$

Ex:- $\frac{100!}{2^n} = \frac{100}{2} + \frac{50}{2} + \frac{25}{2} + \frac{12}{2} + \frac{6}{2} + \frac{3}{2} = 97$

$$100! = 2^{97}\times3^{48}\times5^{24}\times7^{16}\times11^{9}\times13^{7}\times17^{5}\times19^{5}$$

$\frac{100!}{(91)^n} = \frac{100}{13\times7}$ → power of 13 ⇒ Ans = 7

$\frac{100!}{(85)^n} = \frac{100!}{17\times5}$

= ↳ Power of 17 = 50 Ans = 5

$\frac{100!}{(77)^n} = \frac{100!}{11\times7}$ Ans = 9

$\frac{100!}{(65)^n} = \frac{}{13\times5}$

Q1. First 9 multiples of 5 are multiplied then the no. of trailing zero at the ends.

$5 \times 10 \times 15 \ldots\ldots\ldots \times 45$

Cyclicity :-

4 ops

| | For-2 | For-3 | For-7 | For-8 |
|---|---|---|---|---|
| $4n+1 \rightarrow$ Remainder $1 \rightarrow$ | 2 | 3 | 7 | 8 |
| $4n+2 \rightarrow$ Rem $2 \rightarrow$ | 4 | 9 | 9 | 4 |
| $4n+3 \rightarrow$ Rem $3 \rightarrow$ | 8 | 7 | 3 | 2 |
| $4n \rightarrow$ Rem $0 \rightarrow$ | 6 | 1 | 1 | 6 |

$(732)^{227} \rightarrow 2\left(\frac{27}{4}\right) \rightarrow Re \rightarrow 3$

4 → no. of operation

So unit place = 8

$(453)^{272} \rightarrow 2\left(\frac{22}{4}\right) \rightarrow Re \rightarrow 2$

So unit place = 9

2 operation

| For 4 | For 9 |
|---|---|
| 4 → odd | 9 → odd |
| 6 → Even | 1 → Even |

Ex:- $(79)^{91 = odd} = 9$

$(79)^{92 = Even} = 1$

↳ Any power

0, 1, 5, 6 → 1 operations.

-BOD[MAS]

Ex:1

↳ 4-operation

$(76)^{94} \times (22\boxed{7})^{3\left(\frac{43}{4}\right)\ \text{Rem}3} + (7\boxed{1})^{297}$

(Any power; Rem 3)

$6 \times 3 + 1$

$= 19$

$= (9)$ → unit place only.

Work Book / Q-97 / pg-106

GATE 2016/EE

$(2171)^{7} + (2172)^{\frac{9}{4}=1} + (2173)^{\frac{11}{4}=3} + (2174)^{13} \rightarrow \text{odd}$

$1 + 2 + 7 + 4$

$= 14 = (4)$

$(211)^{570} + (146)^{127} \times 3^{\left(\frac{424}{4}\right)\ \text{Rem}=0}$

$1 + 6 \times 1$

$= 7$

# Base system :-

$(25)_{10} = (11001)_2$

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

| 2 | 25 | Rem. |
|---|---|---|
| 2 | 12 | 1 |
| 2 | 6 | 0 |
| 2 | 3 | 0 |
| | 1 | 1 |

$(16+8+0+0+1) = (25)_{10}$

(1) (1)
3 : 24 : 36
2 : 45 : 32
---
6 : 10 : 8

(1) (1) (1)
8 7 3
+ 1 2 7
---
1 0 0 6

NB/pg-96/Q-2

(b=8) 7+6=6+5

(1) (1)
1 3 7
+ 2 7 6
---
4 3 5 $)_{b=8}$

(1) (1)
7 3 1
+ 6 7 2
---
1 6 2 3 $)_{b=8}$

8 borrow 8
7 6 $\not{3}$ 2 1
− . 6 7 2
---
0 3 7 $)_{b=8}$

Ex:-

$$\begin{array}{cccc} 4 & 2 & 2 & 6 \\ +\ 2 & 4 & 4 & 2 \\ \hline 1 & 0 & 0 & 0 & 1 \end{array}_{b=7}$$

If ① ① ①

$$\begin{array}{cccc} 2_1 & 3_2 & 4_3 & 2 \\ -\ 1 & 6 & 5 & 6 \\ \hline 0 & 3 & 5 & 3 \end{array}_{b=7}$$

then 7 7 7

Workbook / Q52 / pg-101

$(7526)_8 - (Y)_8 = (4364)_8$ Find $y = ?$

$$\begin{array}{cccc} 7 & 8_4 & 2 & 6 \\ -\ 4 & 3 & 6 & 4 \\ \hline 3 & 1 & 4 & 2 \end{array}_{b=8}$$

8

Red (3sec) ⟶ 3, 6, 9, 12, ..... 24

Green (4sec) ⟶ 4, 8, 12, ..... 24

K x Lcm (+R1, tG1)

K x Lcm (3, 4) = 12K

R (3t ⟶ 2min) → 120 sec → $\frac{120}{3} = 40$

G (5t ⟶ 3min) → 180 sec. → $\frac{180}{5} = 36$

$$K \times Lcm\begin{pmatrix} R_1 & G_2 \\ 40 & , 36 \end{pmatrix}$$

= K 360

≃ 6 min.

1hr = $\frac{60 \times 60}{360}$ = 10 times