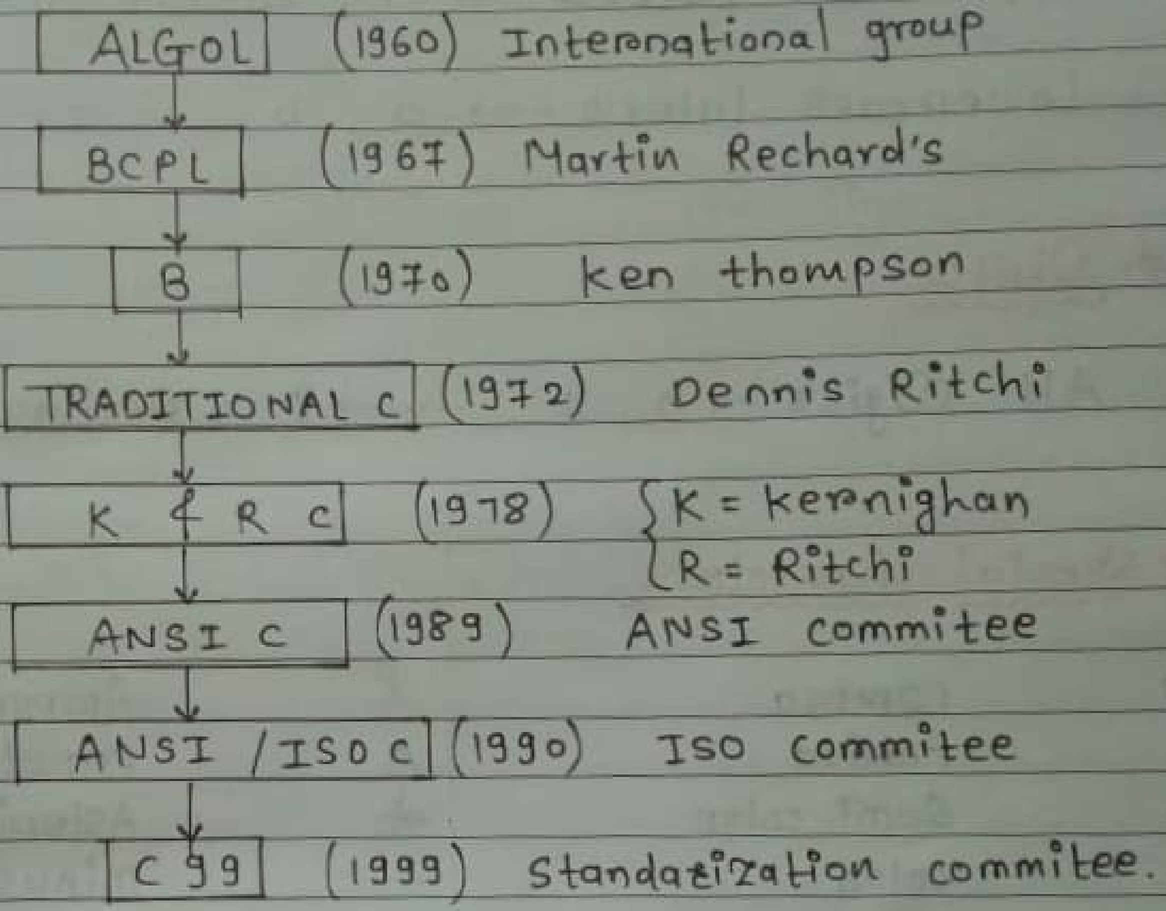
* History of 'c'



ALGOL (1960) International group

BCPL (1967) Martin Rechard's

B (1970) ken thompson

TRADITIONAL C (1972) Dennis Ritchi

K & R C (1978) {K = kernighan, R = Ritchi}

ANSI C (1989) ANSI commitee

ANSI / ISO C (1990) ISO commitee

C 99 (1999) Standatization commitee.

(fig : History of 'c').

* Characteset in c

The character's in 'c' grouped into following four categeris.

1] Later's
2] Digit's
3] Special character's
4] White spaces / Blank spaces.

* Later's

Uppercase later's → A, B ..... Z

Lowercase later's → a, b ..... z

* Digit's

All Digit's → 0, 1, ... 9 (Decimal digit's).

* Special Character's

,	comma	&	Ampersand
.	Period	^	caret
;	Semi-colon	*	Asterisk
:	colon	-	minus sign
?	question mark	+	Plus sign
'	Apostrophe	<	opening angle brac (or less than sign)
"	cotation mark	>	clossing angle bracket (or greater than sig
!	Exclamation mark	(	left parenthesis
|	verticle bar	)	Right parenthesis
/	slash	[	left bracket
\	backslash	]	Right bracket.
~	tilde		
_	underscore		
$	Dollar sign		
%	Percent sign.		

↓

{	left brace
}	Right brace
#	No sign.

* White spaces

Blank space.
Horizontal tab .
New line character.
carriage written.
form feed.

* Trigraph character's

Many non-English keyword's do not support all the character's in that case trigraph sequence is used whic is shown in following table :

Trigraph Sequence	Translation.
??=	# (No sign)
??-	~ (tilde)
??!	| (verticle bar).
??/	\ (backslash)
??\	^ (caret)
??(	[(left bracket)
??)	] (Right bracket)
??<	{ (Left brace)
??>	} (Right brace).

* C Token's

In a passage of text, individual words & punctuation marks are called token's. Similarly in a c programm the smallest individual unit are known as c token's. c has six types of token's. which is shown in the following fig.

C Tokens

Keywords	Identifiers	Constants	String	Special Symbols
eg - int	e.g- main	e.g- 0.2	e.g- "a"	e.g - @
float	add	400	"xyz"	~
if	sum	-7.2	"Purva"	#
while	mul		"1234"	*
break	amount.			{ }
auto				( )

operator's

e.g → +
-
%
/
&
|
|
||
||
<<
>>

* keyword's and Identifiers

Every C word Classified as either keyword or an Identifier. following table shows the keywords in C

2 i] auto
break
case
char
const
continue
do
default

double
else
extern
enum
float
for
go to
if

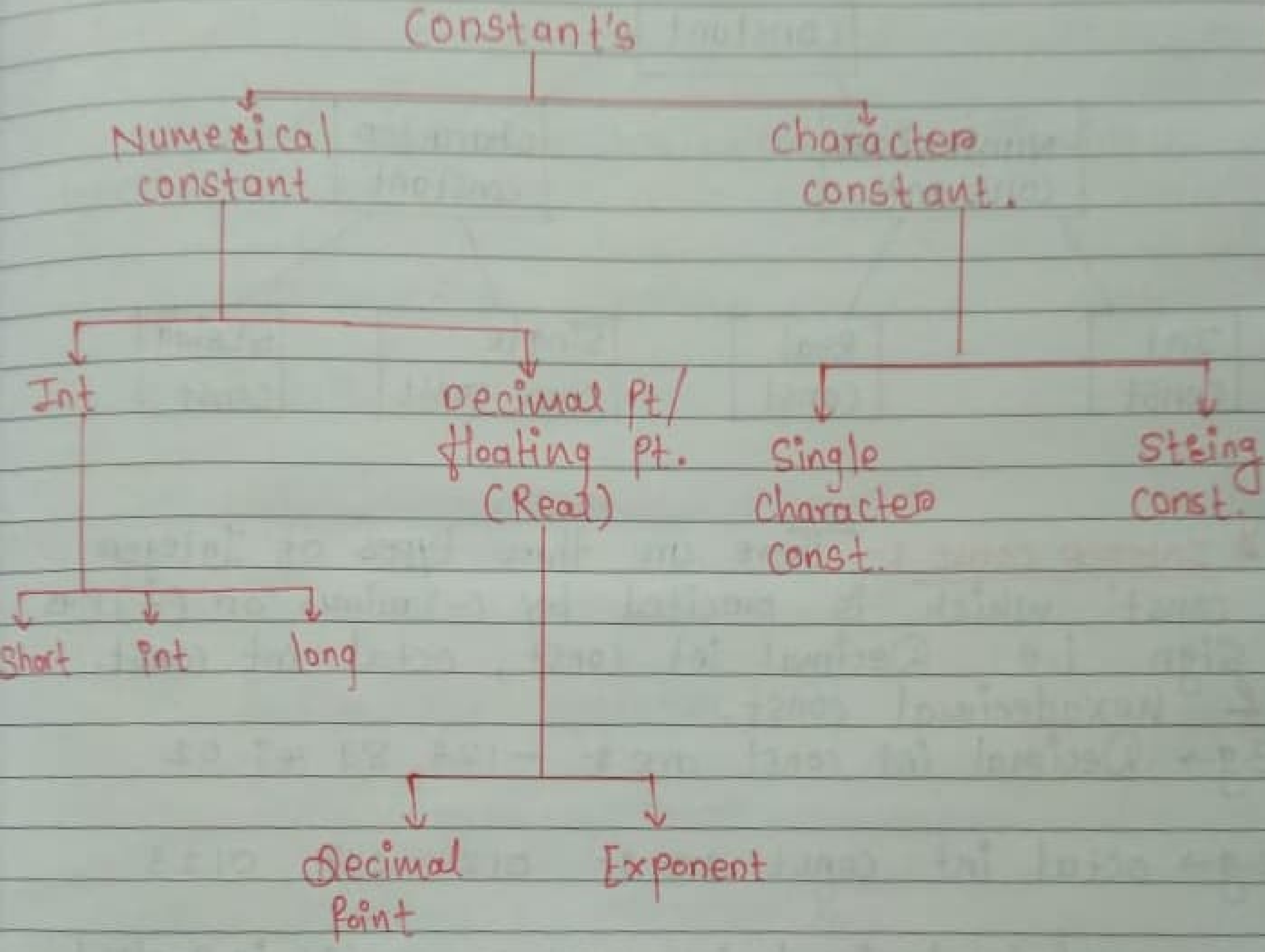
int
long
Register
return
Short
switch
static
signed
sizeof

Struct
typedef.
unsigned
union
void
Volatile
while

There are 32 C keyword's & it must be written in lowercase

Rule :- Identifier contain only the having length up to 31 characters

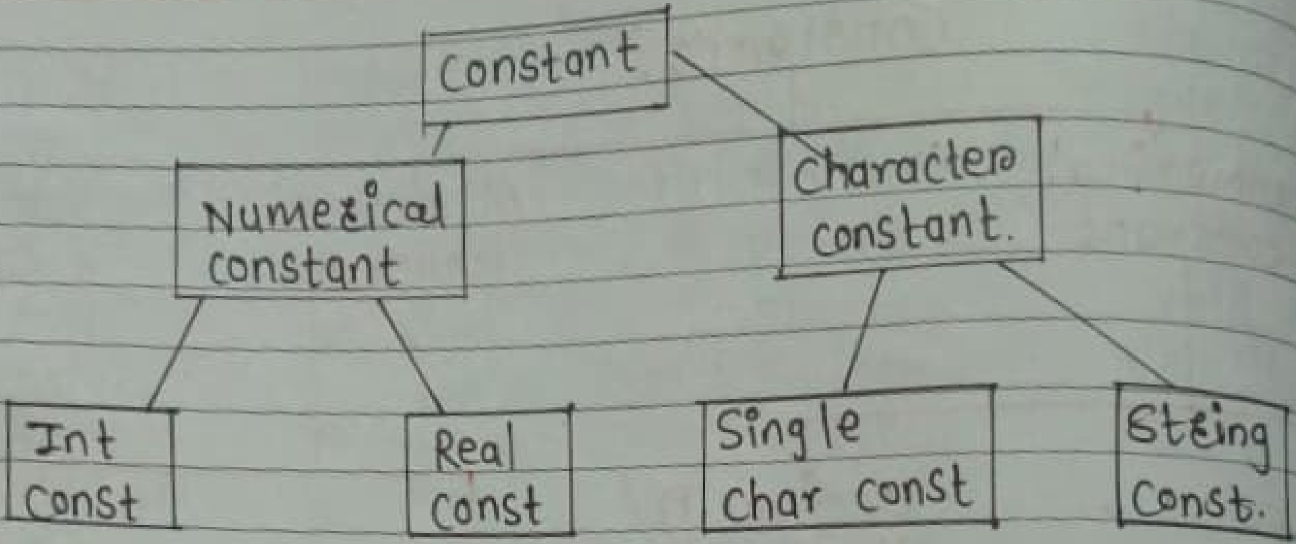
Output ज्या Screen वर print होतो त्याला 'dos prompt' म्हणतात.



* $\begin{cases} e+1 = 10^{1} \\ e-1 = 10^{-1} \end{cases}$

* Stdio → standred input output.
* Conio → consol input output.

* (#) नी start होणाऱ्या statement ला Preprocessor Directive stament म्हणतात.



* Integer const :- There are three types of Integer const which is precited by (-) minus or (+) pl sign i.e Decimal int const, octal int const. & hexadecimal const.

e.g→ Decimal int const are :- −123 89 47 02

e.g→ octal int const are :- 012 017 0123

e.g→ (0x) hexadecimal int const are :- (It is Precite by 0x).

0X123 0x257 0xA253.

* Alt-F5 this key gives output without getch (li

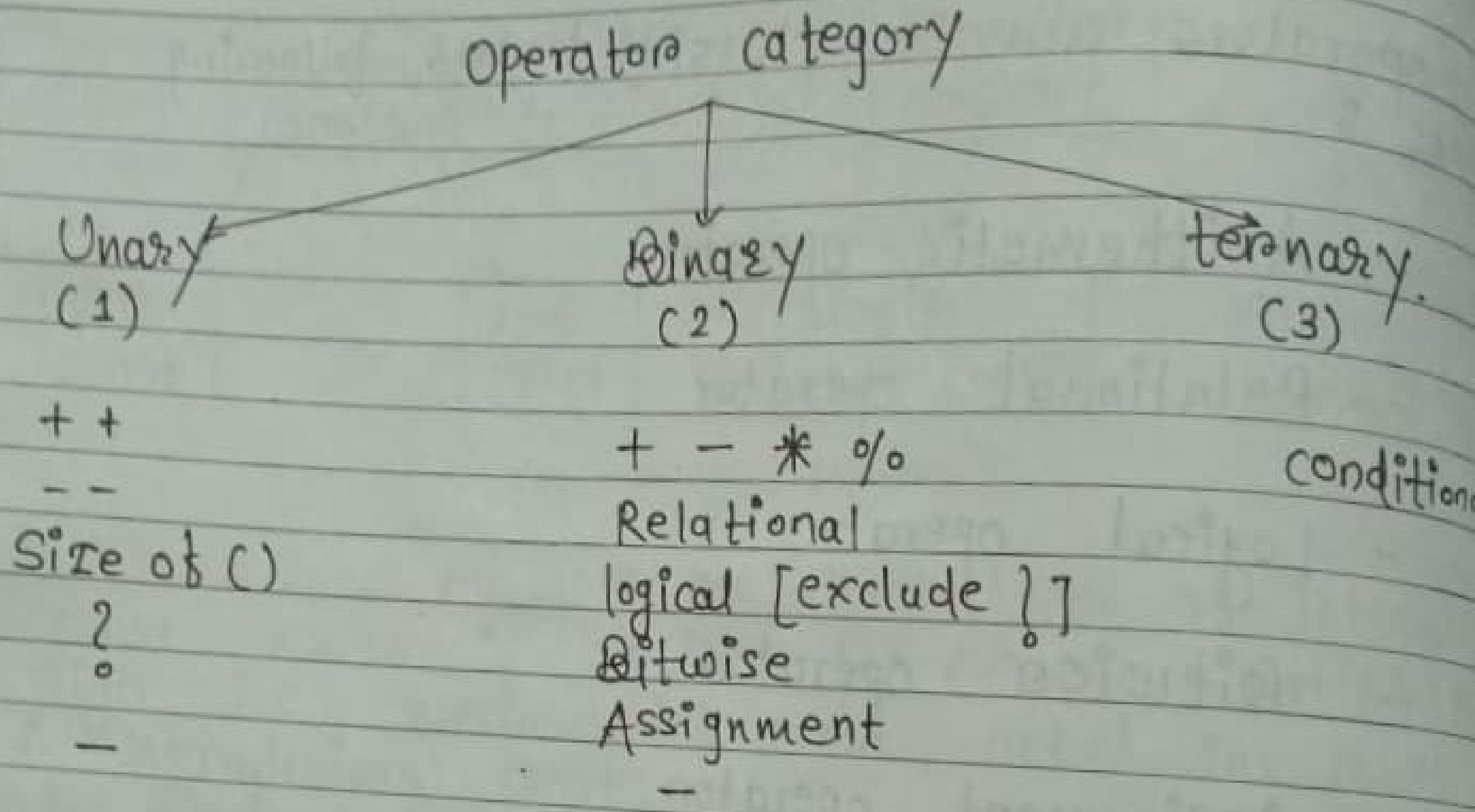
* Operator's in C language.

C operators can be classified into following types :

- Arithemetic operator
- Relational operator
- Logical operator
- Bitwise operator
- Assignment operator
- conditional operator
- Increament operator }
- Decreament operator. } → Special operator.
- Special operator / size of operator.

operator perform an operation on operand.

* Operator Divide into three category.



(-)minus operator are comman for Unary & Binary operator.

(%) Modulo operator can not perfor operation on Decimal no.

(%) Modulo operator '<u>float</u>' वर Perform होत नाहि.

Whenevero Datatype operation perfoem on int type (both) then ans is in int it can not be a float. e.g→

```
int a = 2;
int b = 3;
float c;
```

It is not valid

e.g→

```
int a = 2;
int b = 3;
int c;
```

It is valid.

OR'

Division operation perfoem करतांना दोन पैकी 1 float पाहिजे. e.g →

```
int a = 2;
float b = 2;
float c;
c = a/b.
```

Modulo provide the pemaindero.

e.g →

```
               0  → (Quotient)
Divisoro ← 3 ) 2  → Divisoro
              -0
              ---
               2  → Remaindero.
```

* <u>Program.</u>

```
    # include <stdio.h>
    # include <conio.h>
       void main ()
    {
       clrscr ();
   printf (" %d %d %d \n", 1234, 222222,
                               327668);
   printf(" %d %ld %ld \n", 1234, -222222,
                               -327668 ul);
    getch ();

    }.
```

0 — 9 → askii value

a - 97 → askii value
z - 122

A → 65
Z → 90

(%d) are format specifiers & तो memory मधले
component display. करतो.

जर Program मध्ये only ("%d") given असेल तर
output '0' (zero) येतो becoze zero is the
Smallest no.

(' ') (single ~~coat~~ quote
character)

* Escape Sequence

↓

Skip

Escape Sequence (\) backslash.
which will skip the character which is known to compiler.

\?	→	question mark.
\r	→	return character.
\n	→	newline character.
\a	→	audioble character.
\f	→	form feed.

*** Real constant :-**
The no containg fractional part called as real no OR floating point no OR Decimal Point no e.g→ 17.22, 0.123, -0.2, -8.4 -.8
A Real no also express in (e) notation OR also called as Scientific notation.
e.g→ The value 218.92 will be written as 2.1892e+2 here e+2 means (10^2).

Point RHS ला गेला तर e - no yeto.

Point LHS ला गेला तर e + no yeto.

The format of expⁿ notation would be
mantisa e exponent

Constant	Valid	Remark
698354 L	Yes	Represent the long int no.
25,000	No	comma is not allow in numerical const.
+5.0E3	Yes	valid Exponent const.
3.5e-5	yes	Valid Exponent const.
8.4e 5	No	Because white space is not allowed.
-7.1e-2	yes	valid Exp const.
9.2e2.5 $(10^{2.5})\neq$	No	Exp part cannot be a floating point no it must be an int.
$ 213	No	Special symbol is not allowed.
$ 700	No	Special Symbol is not allowed.
0x	Yes	It is hexadecimal int const.

Any numerical constant doesn't contain special character /symbol.

0X नी No start होत असेल तर No valid आहे
because 0X (hexadecimal) - 0...... 9
A........ f
&
(0x [small] 0X (Cpital))
is valid.

for hexadecimal no 0X is compulsory.
e.g → 0X A110

* Single character constant

A single char const contain a single char which is enclosed within the pair of single quote mark
e.g → 'M' 'a' ';' ' ' the char const has an int value called as ASCII value. The statement printf ("%d" 'a'); will print the askii value of character 'a' i.e '97' And the statement printf ("%c", 'a'); will print the Char a .

* String constants

A string character constant is a Sequence of char which is enclosed within double quotes (" "). following example Show's the string const.

" ", "a", "1987", "....?", "ABC".

* Back Slash char Const

In C There is a Some char with Back Slash which is called as Escape Sequence following table shows the list of back Slash char const.

#tab→ collection of four space ----

variable name maybe uppercase or lowercase and group of char.

Constant	Meaning
'\a'	audioble alert (Bell)
'\b'	back space
'\f'	form feed
'\n'	new line
'\r'	carriage return
'\t'	Horizontal tab
'\v'	verticle tab
'\''	Single quote
'\"'	double quote
'\?'	question mark
'\\'	backslash
'\0'	null.

Datatype is keyword which will define type of data or input which is given by user.

* Variables

A variable is Data name which is to be used to stored a data value variable may take different value during to excution program.

variable name consist of later's Digits and underscore character which is follow the following condition.

1] Variable name begins with later or underscore

2] The length of variable name should be 31 cha

3] Uppercase and lowercase are significant.

4] Variable name should not be a keyword

5] Variable name does not contain white space.

* Datatypes

There are three classes of datatype

1] Primary Datatype

2] Derived Datatype

3] Userdefine Datatype.

* Primary Datatype

Datatype	Range ob value
char	−128 to 127
int	- 32768 to 32767
float	3.4e-38 to 3.4e+38
double	1.7e-308 to 1.7e+308.

* Integer type

Short int	1 byte
int	2 byte
long int	4 byte

Short and long this are use only for int keyword.

* floating point type

float

double

long double

* Characteo types

- The qualifieo signed oo unsigned may be Explicitely applied to char.

- Unsigned char have the value betn 0 to 255 & signed Chao have the value betn -128 to 127

following table shows the Datatype size & Range

Types	size (bit's)	Range
char or signed char	8 bits	-128 to 127
Unsigned Char	8 bits	0 to 255
int oosigned int	16 bits	-32768 to 32767
Unsigned int	16 bits	0 to 65535
short int oo signed short int	8 bits	-128 to 127
Unsigned short int	8 bits	0 to 255
long int oo signed long int	32 bits	-2147483648 to 2147483647

Unsigned long int	32 bits	0 to 4294967296
float	32 bits	3.4e-38 to 3.4e+38
double	64 bits	1.7e-308 to 1.7e+308
long double	80 bits	3.4e-4932 to 1.1e+4932.

* User Define Datatype

There are 2 keyword's are generally used to define user define datatype i.e typedef and enum typedef means "typedefination" and enum means "enamulated datatype"

* Derived datatype

There are some derived datatypes which is derived from primary datatype. C support's arrays, Structures functions, Pointer Derived datatype.

rule of associative

g = 2/2 + 2 * 4/2 - 2 + 2.5/3 =

Where g is of typedef float find the value of g.

* Declation of variable

Variable can be Declared for two things
1] It tell's the compiler show variable name is
2] It Secifited what type of data.

The general format of variable declaration is datatype compiler

Syntax

```
    Datatype    variable name;
for e.g →
            int a;
            float x, y;
```

* Declaration of storage classes

Variable in C cannot hare Datatype but also Storage class that provide the information about location and visibility of the variable There are four storage classes in 'C' namely auto, Static, extern & Register.

following table show's storage & their meaning.

Storage	Meaning
1] auto	local variable know to the fun^n in which it is decleared Default is auto.
2] static	local variable which exist's & retain it's value even after controll is transfer to the calling fun^n.
3] Extern	Global variable known to all fun^n in the program.
4] Register	local variable which stored in Register.

* Assigning the value to the variables

Value can be assign to the variable by using to the assignment operator (=) as follows.

Variable_name = Value

e.g → 1] int a;
a= 7

2] Char C = 'A'

①Defining symbolic const

A const is defeine as follow's by using #define statement.

#define symbolic-name value-const.

for e.g→ #define PI 3.14

#define max 200

ditine MIN 100

following table show's the example of #define statement

Statement	Validity	Remark.
#define sq = 5.7	Invalid	(=) is not allowed.
#define A 3;	Invalid	Semiclon is not allowed.
# define	Invalid	White space is not allowed bet^n # & define
#define a 44	valid	
#define Max 47		
#define a4	Invalid	space is needed in bet^n symbolic const name & it's value.
#define A99, B47	Invalid	only one Symbolic const can be define in single line.
#define PRICE$128	Invalid	($) Dollar is not allowed in const name.

* Operators and Expression.

- Arethmatic operator.
- Relational operator
- Logical operator
- Assignment operator
- Increament / Decreament operator
- conditional operator
- Bitwise operator
- special operator / sizeof operator

* Arethmatic Operator

following table shows the Arethmatic operator & their meaning

Operators	Meaning.
+	Addition or Urnary (+)
−	Substraction of Urnary (−)
*	Multiplication
/	Division
%	Modulo division.

e.g →

If $a=14$ & $b=4$

then

$a+b=18$

$a-b=10$

$a*b=56$

$a/b=3$

$a\%b=2$

Similarly if 1 or both operands of Division operator is -ve then the operation is as follows

$-a/b=-3$

$a/-b=-3$

$-a/-b=3$

Similarly if one or both operands of modulo operator is -ve then Result be. +ve

$-a\%-b=-2$

$-a\%b=-2$

$a\%-b=2$

The sign of Divided is provide to remainder.

```
        n1= n1+n ;

    Printf ("Rererse No is %d", n1);

        if (num = =n1).

    Printf ("Both no are equal: \n");

        if (num = =n1)
    printf ("Both no are equal:\n");

     else
      Printf ("Both no are diffez:\n");

     getch ();
    }
```

OR

```
#include <stdio.h>
   void main () {
    int n, r, mum ;
    long int n1=10;

   long int m=10000L;
printf ("Enter 5 digit number less or
         equal to 32767:\n");

Scanf ("%d", &n);
```

Program :-

Q.1 Write a program to obtain reverse value/no & determine weather the no is equal or not [for five digit no].

```
# include <stdio.h>
# include <conio.h>
void main ()
{
    int n, r, num
        long int n1=0
printf("Enter 5 digit No. less or equal to
            32767 :\n");
Scanf ("%d", &n);
        num=n;
    r= n%10;
    n1 = n1 + r * 1000;
     n = n /10;

     r= n % 10;
    n1= n1 + r * 100;
     n= n /10;
```

```
num = n;
for (int i=0; i<4; i++)
{

    r = n % 10;
    n1 = n1 + r * m;
    n = n / 10;
    m = m / 10;
  }
```

Q.2 Write a program to find the addition of each input no i.e if input is four digit no i.e 1 2 3 4 then find 1+2+3+4=

```
#include <stdio.h>
void main ()
{
   int n r S=0,

printf ("Enter four digit No: \n");
Scanf (" %d ", &n);
```

```
    r = n % 10;
    s = s + r;
    n = n / 10;

    r = n % 10;
    s = s + r;
    n = n / 10;

for (int i=0; i<3 ; i++)
  {
        r = n % 10;
        s = s + r;
        n = n / 10;

     }

        s = s+n;
  printf ("Addition of digits of no is %d", s);

     getch();
   }
```

Q.3 If a four digit no is input through the keyboard write a program to obtain the Sum of first and last digit of this no.

```
#include <stdio.h>
  void main ()
  {
   int n, a, b

  printf ("Enter four digit No:\n");

  scanf (" %d, &n);

   a= n/1000;

   b= n%10;

   a = a+b;

  printf (" Addition first and last digit
          is : %d ", a);

  getch ();

  }
```

.4 Write a program for if 5 digit no is input through the keyboard write a program to print a new no by adding one to each of it's digit e.g→ if the no is 12391 then the output should be displayed as 23402

```
#include <stdio.h>
    void main ()
  {
  long b=0
   int a;
printf ("Entero any fire digit no:\n");

Scanf (" %d ", &n);

  a = n / 100000L;
  a = a + 1;
  b = b + a * 10,000L;
  n = n % 10,000L;

  a = n / 1000;
  a = a + 1;
  b = b + a * 1000L;
  n = n % 1000;

  a = n / 100;
  b = b + a * 100L;
  n = n % 100;

  a = n / 10 ;
  a = a+1 ;
  b = b + a * 10L
  n = n % 10

  n = n + 1;
  b = b + n;

printf ("No is %d", b

getch ();
}
```

Q.5 A cashier has currency notes of denomination 10, 50, & 100 if the amount to be widrown is input through the keyboard find the total no of currency notes of each denomination's the cashier will have to give the widrower

$$\left\{\begin{array}{l}100\times1=100\\50\times1=50\\10\times2=\underline{20}\end{array}\right.$$

$$170$$

* No System
(use for)

1. Binary → system

2. Decimal → human being

3. octal → System Architecture

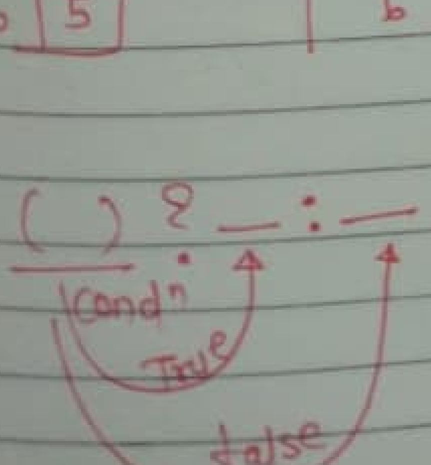
4. Hexadecimal → system Addressing.

a++ → Postfix operation

++a → Prefix operation

int a=5;

b = a++;	b = ++a
a $\boxed{6}$	a $\boxed{6}$
b $\boxed{5}$	b $\boxed{6}$



* O/P

- What will be the output of following program

```
#include <stdio.h>
    void main()
  {

    int i=3;

    i = i++;
  printf("%d", i);
  }
```

~~Minor~~ Manul Trace

i

$\boxed{3}$

i=3

i=3++

i $\boxed{4}$

i++ → Postfix

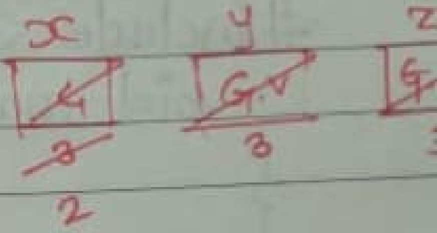
++i → Prefix

Assignment oprator ला effect होतो.

O/P →

i = 4

Find the output of following program.



```
#include<stdio.h>
    void main ()
    {
      int   x=4 ,y, z;
      y = --x
      z = x--;
    printf("%d %d %d",x,y,z);
```

O/P →

2 3 3

x=2 y=3 z=3

- Find the output of following program.

```
#include <stdio.h>
 void main ( )
{
  char ch;

   ch= 'A'

 printf(" The later is");
 printf("%c", ch >= 'A' && ch < ='Z' ?
                  ch +'a' - 'A' : ch);
 printf(" %c", ch>= 'A' && ch < ='Z' ?
                  ch:ch+'a' - 'A');

 }
```

O/P →

what will be the output of following program.

```
#include <stdio.h>
    void main ()
   {
      int i=2;
     int j = i+(1,2,3,4,5);
    printf ("%d", j);
```

O/P →

hat is output of following program.

```
#include <stdio.h>

    void main ()
  {
      int x=55;
  printf("%d %d %d", x<=55,
          x=40, x>=10);

  }
```

$x = \boxed{55}$ assign

55

$\left.\begin{array}{l} x \leqslant 55 \\ x <= 55 \end{array}\right\}$ $\underline{T\ (1)}$

$\underline{x = 40}$

$x >= 10$

$40 >= 10\ \underline{(T)\ 1}$

O/P →

1 40 1

- What is the output of following program.

```
#include <stdio.h>
   void main ()
  {
   int k, num = 30;
  k=(num >5 ? (num <=10 ? 100:200): 500);
  printf ("%d", ~~num~~ k);

  }
```

O/P →

Temprorary operator
bracket.

( ) → nested conditional Statement.

( ) Parenthesis → Having the highest preference in programming language

- What will be the output of following program.

```
#include <stdio.h>
   void main ()
  {
   int a=100, b=200, c;
      c=(a== 100 || b>200);
   printf ("c = %d", c);

   }
```

a $\boxed{100}$ b $\boxed{200}$

Rel^n operator

c = (a == 100 || b > 200

(T)=1 (F)=0

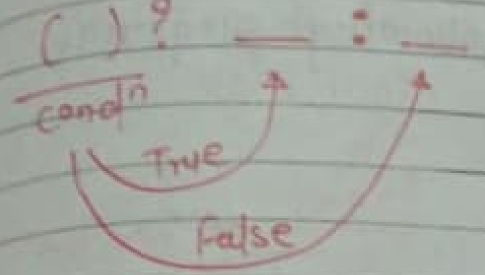
F || F = T

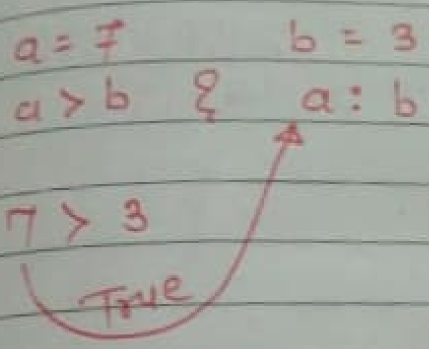
T (1)

O/P → 1

Table

A	B	A&B	A|B	A^B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0





If condn is false then compiler curly brace च्या आत म्हणजे statement excute करत नाही direct बाहेर जाते curly brace च्या.

```
#include <stdio.h>
  void main ( )
 {
   int  x=12 , y=7 , z;
   z = x != 4 || y== 2;
   printf ("z = %d", z);
 }
```

x | y | z
12 | 7 | ~~4~~ 1

z = x != 4 || y== 2

12 ≠ 4   7==2

(T) = 1   ≠

$\boxed{z = 1}$

O/P →

1

```
#include <stdio.h>
  void main ( )
 {
 int, i=4 , j= -1 , k=0 ,w ,x, y, z;
   w = i || j || k;
   x= i&& j&& k;
   y= i || j && k;
   z = i&& j || k;
Printf ("%d %d %d %d", w,x,y,z);
```

i | j | k
4 | -1 | 0

w = i || j || k;

$\boxed{w=1}$

x = i&& j&& k;

$\boxed{x=0}$

y = i || j&& k;

1|| 0&& 1

y = 1&& 1 = 1 $\boxed{y=1}$

z = i&& j || k;

1&&0 || 1   0|| 1

z=1

O/P →

1 0 1 1

False $\xrightarrow{\text{compiler}}$ -ve value

True $\xrightarrow{\text{compiler}}$ +ve value.

<< ← Bitwise left shift operator

>> ← Bitwise Rightshift operator.

* Binary conversion

```
          64  ← Quotent
    → 2 ) 129 ← Dividend
Divisor   12
          009
           -8
            1 ← remainder

          32
      2 ) 64
          -64
          00
```

```
2 | 129     1
2 | 64      0
2 | 32      0
2 | 16      0
2 | 8       0
2 | 4       0
2 | 2       0
    1
```

continue

Till Divisor > Quotent.

1	0	0	0	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

128 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 129

. Find output of following program.

```
#include <stdio.h>
 void main
 {
   int a=2, b=4, c=5
   int x, y, z
   x = a & b;
   y = a | b;
   z = a ^ C;
printf("%d %d %d", x, y, z);
 getch();
 }
```

a $\boxed{2}$ 10

b $\boxed{4}$ 100

c $\boxed{5}$ 101

$$\begin{array}{r} 10 \\ *100 \\ \hline 00 \\ 000 \\ 1000 \\ \hline 1000 \end{array}$$

x $\boxed{8}$ 1000

$$\begin{array}{r} 10 \\ +100 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 010 \\ 101 \\ \hline 111 \end{array}$$

z $\boxed{7}$ 111

y $\boxed{6}$ 110

$2^2\ 2^1\ 2^0$

$2^2\ 2^1\ 2^0$

$1\times2^2+1\times2^1+1$

$4+2+1=7$

$1\times2^2+1\times2^1=0$

$4+2+0=6$

O/P →

8, 6, 7

```
# include <stdio.h>
  void main ()
 {
   int  l = 9 , m = 8 , n = 6;
    int  a , b , c ;

    a = ~a ;
    b = a << 2;
    a = a & l;
    b = a ! b;
    c = ++ l ;
    c = c ^ m;
  printf ("%d %d %d", a, b, c);
    getch ();
 }
```

l $\boxed{10}$ 1010

m $\boxed{8}$ 1000

a $\boxed{}$

b $\boxed{}$

c $\boxed{\cancel{10}\,2}$

$$\begin{array}{r} 1010 \\ \wedge\, 1000 \\ \hline 0010 \end{array}$$

a → $\overline{\text{G.V}}$, b → $\overline{\text{G.V}}$, c → 2

O/P →

G.V , G.V , 2

```
#include <stdio.h>
  void main ( )
  {
                                a
   int a=300 , b, c
   if (a>=400)                [300]
      b = 300;
      c = 200;   F
   printf ("\n %d %d", b, c);
                      G.V  200
  }
```

```
O/P →
      b → G.V
      C → 200
```

```
#include <stdio.h>
   void main ()
   {
      int a= 500, b, c;
       if (a>=400)
        500>=400
                     T
      b = 300;

      c = 200;
   printf("\n%d %d", b, c);
                     300 200
   }
```

```
O/P →
300 200
```

```
#include <stdio.h>
    void main ()
  {
    int x=10, y=20;
     if (x==y);
     printf ("\n %d %d", x, y);
                         10  20
  }
```

O/P →

10 20

```
#include <stdio.h>
    void main ()
  {
    int x=3, y=5;
     if (x==3)
        3==3
                            → 3
     printf ("\n %d", x);         → error
else;
                            → 5
     printf ("\n%d", y);
  }
```

semicolon while असल्यामुळे condn true असो किंवा false statement खाली जाणार नाही तो Print होणार becoz of semicolon

```
#include <stdio.h>
    void main ( )
  {
      int  i = 1 ;
   while  (i <= 10);
             i<=10
    {
       printf (" /n %d", i);

            i ++ ;
       }
  }
```

O/P → 1

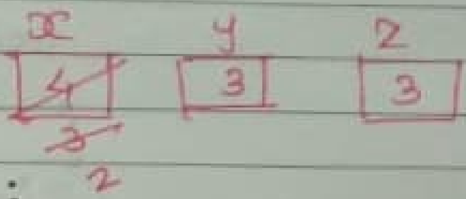
* Blue colour window called as → Editor
* Black colour window called → output window.

```
#include <stdio.h>
     void main ( )
  {
     int  x=4;
   while  (x==1);
             4==1
    {
          x =x-1;
     printf ("/n %d", x);

          --x;
        }
    }
```

O/P →

No output

```
#include <stdio.h>
   void main()
   {
     int x = 4 , y , z ;
       y = --x ;
       z = x-- ;
     printf("/n %d %d %d", x, y, z);
     }
```



```
O/P →
        2 3 3
```

Postfix असेल तर operation आधी Perfor करायचं.

$z = x_{(4)} == -y$

$y = 3$ $\quad -3$

$4 - 3$

$= 1$

$\boxed{z=1}$

```
#include <stdio.h>
   void main
   {
   int x=4 , y=3 , z ;
   z = x--   -y ;
  printf("/n %d %d %d", x, y, z);
   }
```

```
O/P →     3   3   1
```

```
#include <stdio.h>
  void main ()
  {
   int x=4, y=0 , z;

   While (x>=0)
    (-1 0 1 2 3 4 >= 0
    {

       x--; 3 2 1 3 -1

       y++; 1 2 3 4 5

int x = a-b
    5-6    if
  x=-1
          (x==y)

   continue;

   else
   printf("/n %d %d", x, y);
        }

3   1       }
1   3
0   4
-1  5
```

```
o/p →  3   1
       1   3
       0   4
       -1
```

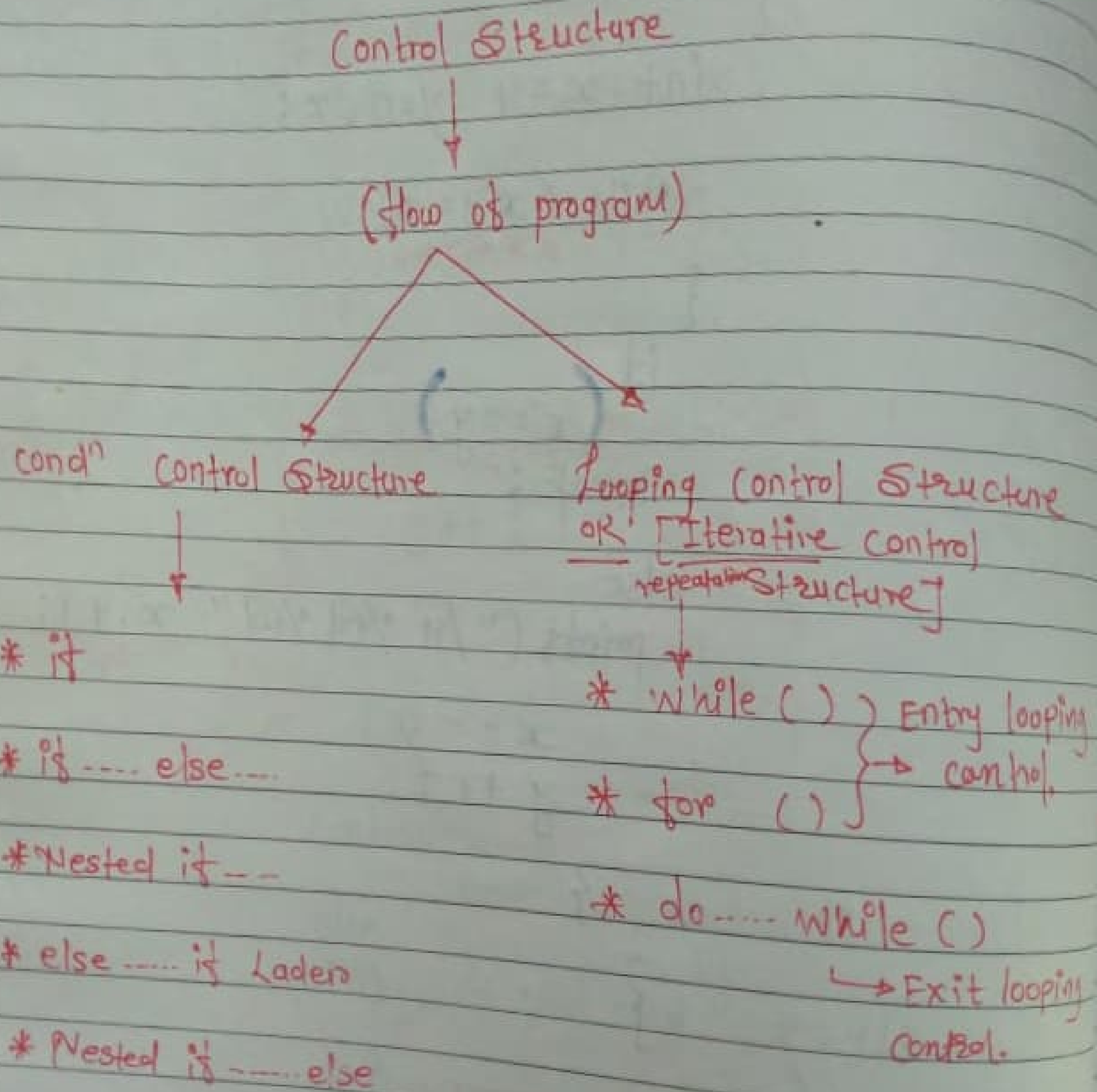
```
# include <stdio.h>
    void main()
  {
    int x=4, y=0, z;

    while (x>= 0)
          2 ~~3~~ ~~4~~ >=0
    {
      if (x==y)
          2==2
      break;

     else
      printf("/n %d %d", x,y);

            x--;
            y++;
         }

      }
```

```
o/p →
        4  0
        3  1
```

*Iterative = Repeatation

* Control Structures

The steucture which control the folow of program



Control Structure

(flow of program)

condn Control Structure

* if

* if ---- else ---

* Nested if ---

* else ----- if Ladder

* Nested if ------ else

Looping control Structure

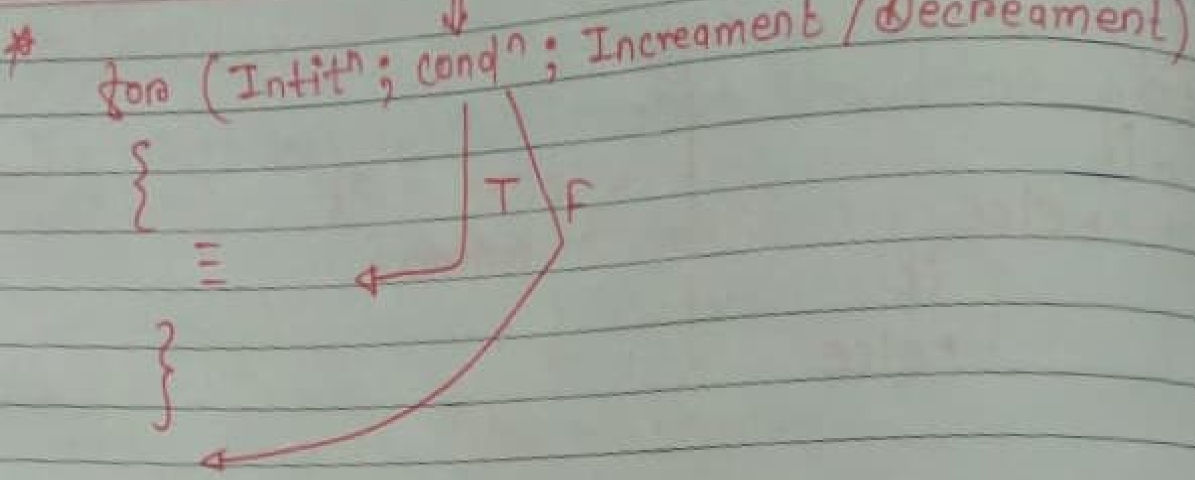
OR [Iterative control repeatation Structure]

* While ( )
* for ( )

→ Entry looping control.

* do ----- While ( )

→ Exit looping Control.

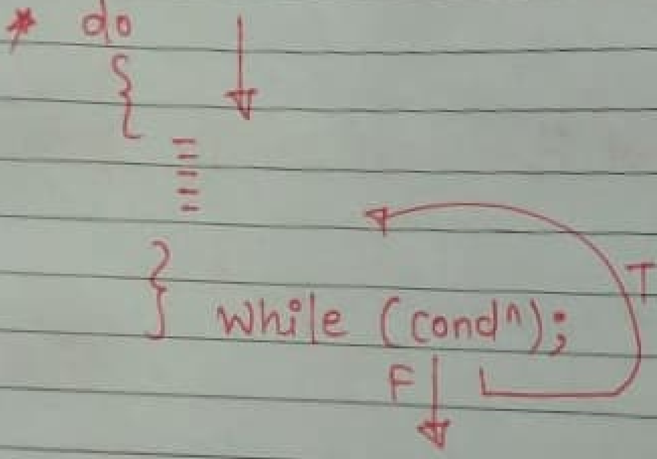


* for (Intit^n; cond^n; Increament/Decreament)

{

≡

}



* do

{

≡

} while (cond^n);

*

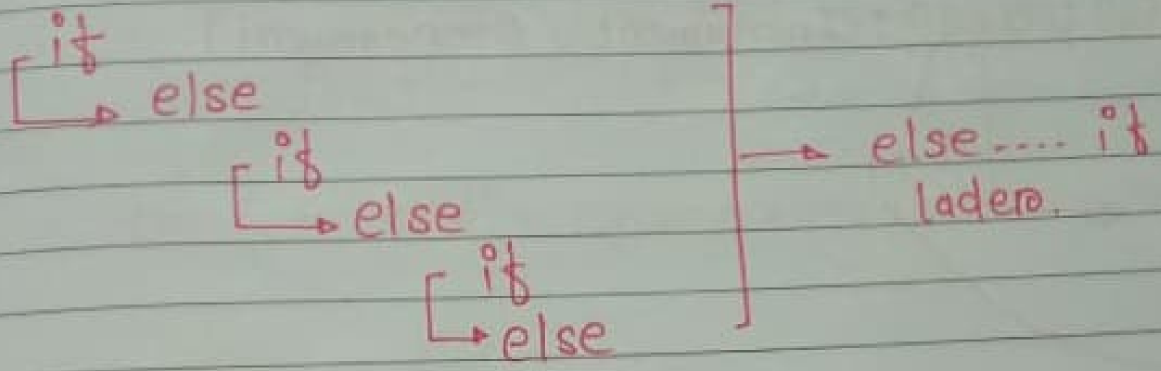
for (Init^n; cond^n; Increament/Decreament)

{

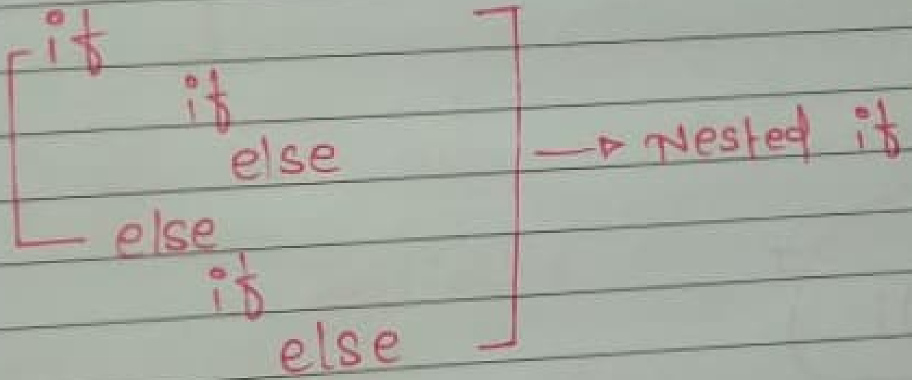
≡

}

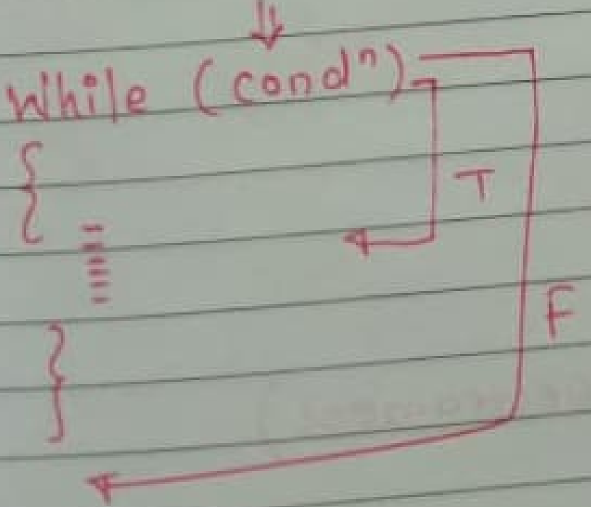
*


if
else
if
else
if
else
else.... if
ladero.

*


if
if
else
else
if
else
Nested if

*


While (condn)
{
}
T
F

OR

```
    Initial$^n$;
for (; cond$^n$; Increament/Decreament)
  {

  }
```

OR

```
      Initial$^n$;
  for (; Cond$^n$;)
  {
        Increament/Decreament;

    }
```

*

Eg →

```
for (int x=0 ; x<5 ; x++)
  {
    ⋮
  }
```

OR

```
int x=0 ;
for ( ; x<5 ; x++)
 {
   ⋮
 }
```

OR

```
int x=0 ;
for ( ; x<5 ; )
 {
          x++ ;
 }
```

```
                                              0 1 2 3   5 4 3 2
*    for (int x=0, y=5 ; x < y; x++, y--)
                  0      5    3 2 1 0 < 5 4 3 2
     {
             printf("%d", x);

       }
```

```
0
1
2
```

Note's

* If in for loop if more than one variable is used then it is seperated by (,) comma
  oR
  combine

* And for condition use the logical and (&&) and logical or (||) compulsory.

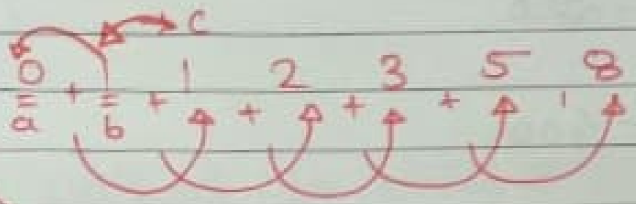
* Program

```
#include <stdio.h>
#include <conio.h>
   void main ( )
   {

       int a=0 , b=1, c n;

       printf("How many no you want to
             print for fiboncii Series :\n");

       Scanf("%d \t %d \t", a, b);

         for (int i=3 ; i<=n ; i++)
         {
              c = a+b
              printf("%d \t", c);

                a=b ;
                b=c ;
              }

              getch ( );

         }
```

→ logic o
program

n	a	b	c
5	0	1	1
	~~1~~	~~1~~	~~2~~
	~~1~~	~~2~~	3
	~~2~~	~~3~~	5
	3	5	.

0 1 1 2 3 5 8



c = a + b
1 = 0 + 1
2 = 1 + 1
3 = 1 + 2

O/P →
How many no you want to print for fiboncii Series
7

* An electtic power distribution company Charges it domastic consumer as follows

consumption unit	Rate of Charge.
0 – 200	Rs 0.50 / unit
201 - 400	Rs 100 plus Rs 0.65 / unit excess of 200.
401 - 600	Rs 230 plus Rs 0.80 / unit excess of 400.
601 and abore	Rs 390 plus Rs 1.00 / unit excess of 600

→

```
# include <stdio.h>
# include <conio.h>
    void main
  {
     int u, cn;
     float b,

    printf(" Enter unit of electricity &
              costomer no : \n");

    scanf ("%d %d", &u, &cn);
     if (u<=200)

          b= u * 0.50;

     else if (u>=209 && u<=400)

              b= 100 + (u-200)* 0.65 ;

     else if (u>401 && u<=600)

              b= 230+ (u-400) * 0.80;
      else
              b= 390 + (u-600) * 1.00;

     printf ("customer no is: %d \t
               Billing is : %f \n", cn, b);
    getch ();
  }
```

&& → Compulsory use

|| → optional use.

(T OR F)

If cond^n is not confirmed then use nested if else

If desigin होत cond^n check करायच असेल तर use Lader else if

cond^n check करुन double double cond^n check करायचा असेल तर looping use होतो.