2 - 04-2024 INCREMENT & DECREMENT :- ++, -- [UMARY] There are two notations called 1. PREFIX : b=++a a= A+1, b= A Here a will observe 2. Postfix: b= a++ Change to [+1]+/++a only & will not change b= a, a= a+1 in both cases b=6, b=5 Ex of Prefix: -> main () 0/9/-1 int a=8,6; b=++a; point ("=1. d. d. d", a, b); in the first trainst. Ex of Postfin: --> main() O/P: - N=9 int u=8,6; b= a++; fruitf ("1.0".0", a, b); -> main () 0/p ;- 12 21. 1 int a=1, b=2, c=3, d; d=++a+b++++c; printf ("1. d 1. d 1. d 1. d", a, b, c, d) 1=1.0, b=60, c=60, d=1.0

-> mais () 11-2 2 ind a=1, b=2, C=3; d; 6=43 d=++a+b+++c++; want (" y. d 1, d 1, d 1, d", a,b, c,d); -> main () I int u=1, b=2, (=3, d) 1/p! - 12=12 d = a++ + b++ + c++) printf ("1. d 1. d 1. d 1. d", a, b, c, d); -> main() 2 ×3 × 4 { int a=1, d; d = a++ + a++ + a++; printf ("1. d 1. d", a, d); > main () 2 int u=1, b; b=+Fa+++a+a++ printf (" 1. d", b) -> main () { int a=1; point ("1. Q", a);

```
-> main ()
  1 int a = 1, b;
                          0/1:- b=1
    b = a++',
    printf ("bt. d", b);
                              the sequence of the second
               when the other they is the best of
* We shouldn't shouldn't apply inverment & devement
   operators on constants
                      7 It sould supplied that a confe
 DEED DECREMENT OPERATORS =>
-> main ()
  · f int. a= 5, b;
    b= A -- ;
     printf ("a=1. d, d", a, b);
                                       Christian.
> main ()
                                  3 40 3 3 40 1
   1 int a 25,6;
                                  11: - 100 m
    b= -- A;
     paintly ('i'l. d, b', a, b);
                            and the land large
9 main ()
  f int a = 5, b;
                                    0/0;-
    5= ++ a + a++ + -- a + a -- ;
```

29-04-2024 RECATIONAL: - L, 2, L=, >=,==,!=1.1. ite sty with an BINARY OFFIATORS SYNTAX: - oful Rop oful These operators where the relation between 2 operands. If it is true gives a true value means 1, otherwise gives a falle Value means o? 0/9:-1 Ex main () ! int a; A= 5 >= 2; } paint ("1.0", a); Charles & Start main () 0/p : - 0 1 man L int a' 195 V 14 A = 5==2) trans ["7.0", a); Logical And & & Chinary operator) EX: LOGICAL :-A B A A A B (inta;" main () A=566kk8>5; print (" ", d", a);

egical OR 11 (binary operation) logical Not! (Unary) A B AIIB Eximain () Eximum () tint a; t int a; M252611625) a=5k45; founds ("1.d", a); fruit (" 1. 0" 1); 3 0/p:-1 \$ 0/p: +1 Brighain () Ex: main () int a; Linta) 11 0 a=6!=5; W=15;12: 1 punily (" 1.d", a); wantf 1" 7. d", a); 5 0/0:-1 BIT WISE OPERATOR => we should not apply & situise And & (binary) Bituise operators on float bituis on 1 (binary) bithing xor ^ (can binary) 1's complement ~ (talda) (klmary) left shift (< (binary) right shift >> (binary)

EX', 0/p 17 muin () { int a, b, c', - INDIVIDUAL OPERATION A=5 X6; b= 516; C= 516; want (" 1. d 7. d 7. d", a, b, (); Ex C. Maria 8-10.00 main () 12=1100 Lind a, b, c, d, e. d= 1000 38 Az 8 4 12; 11=1100 >11 b = 8 11 12 ; tons C= 8 x 12 ; 12 0100 ->4 A=8.112; e= 8 12; paintf ("1.0".0".0.1.0", a, b, c, d, e); M. Appliance = 3 while they bloom, and 1 1 8 12 4 (provided & son would be Cotton Coi 1-1 Employ TRUTH TABLE OF THE BLTWISE OPERATORS = > X AY X I Y X AY O TO TO THE O Committee that the Company of that had

LEFT SHIFT OPERATOR !-A << n = 1 x 2 m -> main () f int a=5, b; 0/p: - 5 # 23 2 40 [ceftshift] b= A < < 3; EOF - L CRIPH SHIFTS pointf (" 1. d", b); 5/23=0 RIGHT SHIFT OPERATOR "- $A >> M = \frac{A}{2^m}$ -> main () 01P: - 90 50 = 66 [Rylushift] { int 11250, 6; b = a>> 3; Left shift:wintf (" 7. 1", b); 1. 50t 23 = 400 19 = 9 LEFT SHIFT & RIGHT SHIFT OPERATORS => -> main () 2 int n; frintf (" enter the value of n: "); scanf ("7.d"; tx); fruitf (" Left shift operation: \m x << 1 = 1.d\m", x << 1); fruit | "Right shift operation: 1 m. x >> 1 = 1. d 1 m", x >> 1); enter the value of x: 19 left shift operation: 19/21 = 9 Right White Operation.

X>>1=

```
1 Q:- To demonstrate the use of bituise operations:
  -> main ()
     int a, b;
       printy ("Enter the value of a: ");
       scant ("7.d", ta);
       brandf (" Enter the Value of b: ");
        scant ("1.d", 4b);
       pointf (" The Bituis operations were as helpiv: 1 m 1).
        part (" a b = 1. d \ m", a b b);
        fruit ("alb = 1.d \m", alb);
         founty ("a'b= 1.d \m", a'b)
         fountf ("a<<1=7.d(m"), a<<1))
         foundf ("a>>1 = 1.0 (m", d>>1);
         ("b<<1= 1.d \m" b<<1)
         fount ("b>> 1= 11. din', b>>1);
  0/p; -
                                          ACCD=AX2
    Enter the value of a: 5
                                          a>>b= a/2b
     Enter the Value of 6: 9
                                        b < (a = b \times 2^{a})
b > > a = \frac{b}{2^{a}}
b = b = b
   The Bituire operations are as helow
                                         MAB = 1
    A4b=1
                                         a 16 = 13
    alb= 13
                       his known products
                                         1 b = 12
    A b = 12
                                         A < < b = 25 60=>5x1
    A << 1 = 10
                                         A >> b = 0 => 5/21:11
    a >> 1= 2
                         To the second of the
                                         b < < 1 = 288=>9x1
                         with the second
                                         b > > 1 = 0 => 9 = 1 |
     P <<1=18
                          and the ship of
     6 >> 124
```

```
BITWISE OPERATIONS :-
                         main ()
                         lint a, b;
                           pointf ("Enter the Value of a: ");
                            scanf ("1.0", 4 a);
                            jointly (" Enter the value of b: ");
                            scanf ("7.8", 46)
                            printf ("The Bituise operations are as below: 1 m");
                             printf (" a4b = 1.d(m", a4b); 11 Rituire AND operation
we as lielow: (m").
                            jount (" alb= 1.d \n", alb); 1/Bituise or operation
                             print ("Allb= 7.d (m"), allb); 1/ logical or operation
kb);
                             print ("1 1 b = 1 d m", 1 b); "Bituire xor operation
16);
                            pountf (" a < < b = 1. d \ m'), a < < b); 1/Bituil left-shift operation
16)
                             fountf (" acc 1= 1.dim", acc 1);
                                                                     1<16= 1x(21b)
1 ((1)
                            printf ("a>>b= 1.d \m'; a>>b); //Bituin right-shift operation
a>>1)
b < < 1);
                                                                   1>>b= 1/(2/3)
                            built ("A>>1= 1.dim", a>>1);
PITTY,
                            printf ("b<<a>1.0(m"), b<<a>1); // Bituine left-shift operation</a>
                                                                  b((4: b* (2^u)
                            built (" b << 1= 1.0 (m", b << 1);
   ACCh=AX2
                            foundf ("b>>a='/.dim'), b>>a); // Bituine Right-Shift operation
                                                                 b>>a= b/(2^a)
                            foundf ("b >>1= 1. d\m", b>>1);
  6>>A - b
  ad b = 1
  1 1 1 2 13
  1 b = 12
  A < < b = 2560=>5x2
  A>> b= 0 => 5/21=1
  bec d= 288=>4x2
  b>> A = 0 => 9 = 0.11
```