Operators: Operator is the symbol which performs some operations on the operands.

Types of operators

1.Arithmetic operator (+ - * % /)

Operator	Description	Example (a=4 and b=2)
+	Addition of two operands	a + b = 6
-	Subtraction of two operands	a – b = 2
*	Multiplication of two operands	a * b = 8
/	Division of two operands	a / b = 2
%	Modulus gives the remainder after division of two operands	a % b = 0

//Arithmetic operator (+ - * % /)

```
#include<stdio.h>
int main()
{
        int a,b,add,sub,mul,div,mod;
        printf("Enter two value:");
        scanf("%d%d",&a,&b);
        add=a+b;
        sub=a-b;
        mul=a*b;
        div=a/b;
        mod=a%b;
        printf("\n sum is:%d",add);
        printf("\n subtraction is:%d",sub);
        printf("\n multiplication is:%d",mul);
        printf("\n division is:%d",div);
```

```
	printf("\n modulus is:%d",mod);
	return 0;
}
```

Output:

```
	Enter two value:100 200

 sum is:300
 subtraction is:-100
 multiplication is:20000
 division is:0
 modulus is:100
```

```
//Arithmetic operator ( + - * % /)
#include<stdio.h>
int main()
{
	int a,b;
	printf("Enter two value:");
	scanf("%d%d",&a,&b);

	printf("\n sum is:%d",a+b);
	printf("\n subtraction is:%d",a-b);
	printf("\n multiplication is:%d",a*b);
	printf("\n division is:%d",a/b);
	printf("\n modulus is:%d",a%b);
	return 0;
}
```

Output:

```
	Enter two value:100 200

 sum is:300
 subtraction is:-100
 multiplication is:20000
 division is:0
 modulus is:100
```

```
//Arithmetic operator ( + - * % /)
#include<stdio.h>
int main()
{
	int a,b,add,sub,mul,div,mod;
```

```
        printf("Enter two value:");
        scanf("%d%d",&a,&b);
        add=a+b;
        sub=a-b;
        mul=a*b;
        div=a/b;
        mod=a%b;
        printf("\n sum is:%d \n sub is:%d  \n mul is :%d \n div is :%d \n
mod is:%d",add,sub,mul,div,mod);

        return 0;
}
```

Output:

```
Enter two value:100 500

 sum is:600
 sub is:-400
 mul is :50000
 div is :0
 mod is:100
```

2.Unary operator(+ - ++ --)

Operator	Description	Example(count=1)
+	unary plus is used to show positive value	+count; value is 1
-	unary minus negates the value of operand	-count; value is -1
++	Increment operator is used to increase the operand value by 1	++count; value is 2 count++; value is 2

--	Decrement operator is used to decrease the operand value by 1	--count; value is 1 count--; value is 1

//Unary operator(+ - ++ --)

```
#include<stdio.h>
int main()
{
	int a;
	printf("enter the value:");
	scanf("%d",&a);
	printf("+a value is:%d",+a);
	printf("\n -a value is:%d",-a);
	printf("\n ++a value is:%d",++a);
	printf("\n a++ value is:%d",a++);
	printf("\n a++ value is:%d",a);
	printf("\n --a value is:%d",--a);
	printf("\n a-- value is:%d",a--);
	printf("\n a-- value is:%d",a);
}
```

Output:

```
enter the value:100
+a value is:100
 -a value is:-100
 ++a value is:101
 a++ value is:101
 a++ value is:102
 --a value is:101
 a-- value is:101
 a-- value is:100
```

3.Relational operator(< <= > >= == !=)

Operator	**Description**	**Example (a=10 and b=20)**

<	less than, checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) value is 1(true)
<=	less than or equal to, checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) value is 1 (true).
>	greater than, checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a > b) value is 0 (not true).
>=	greater than or equal to, checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) value is 0 (true).
==	equality ,checks if the value of two operands is equal or not, if yes then condition becomes true.	(a == b) value is 0 (not true).
!=	inequality, checks if the value of two operands is equal or not, if values are not equal then condition becomes true	(a != b) value is 1 (true).

```
//Relational  operator(< <= > >= == !=)
#include<stdio.h>
int main()
{
        int a,b;
        printf("Enter values:");
        scanf("%d%d",&a,&b);
        printf("\n a<b value is :%d",a<b);
        printf("\n a<=b value is :%d",a<=b);
        printf("\n a>b value is :%d",a>b);
        printf("\n a>=b value is :%d",a>=b);
        printf("\n a==b value is :%d",a==b);
        printf("\n a!=b value is :%d",a!=b);
}
```
Output:

```
Enter values:100 200

 a<b value is :1
 a<=b value is :1
 a>b value is :0
 a>=b value is :0
 a==b value is :0
 a!=b value is :1
```

4.Logical operator(&& || !)

Operator	Description	Example
&&	Logical AND operator. If both the operands are true then condition becomes true.	(5>3 && 5<10) value is 1 (true).
||	Logical OR Operator. If any of the two operands is true then condition becomes true.	(5>3 || 5<2) value is 1 (true).
!	Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(8==8) value is 0 (false).

```
//Logical  operator(&& || !)
#include<stdio.h>
int main()
{
	int a,b,c,d;
	printf("Enter values:");
	scanf("%d%d%d%d",&a,&b,&c,&d);
	printf("\n a<b && a>c value is :%d",a<b && a>c);
	printf("\n a<b || a>c value is :%d",a<b || a>c);
	printf("\n !(a==b) value is :%d",!(a==b));
```

}

Output:

```
Enter values:100 200 250 300

 a<b && a>c value is :0
 a<b || a>c value is :1
 !(a==b) value is :1
```

5.Assignment operator(+=,-=,*=,/=,%=,>>=,<<=,&=,|=,^=)

Operator	Description	Example(a=4 and b=2)
+=	a=a+b	a+=b; a=a+b = 6
-=	a=a-b	a-=b; a=a-b = 2
*=	a=a*b	a*=b; a=a*b = 8
/=	a=a/b	a/=b; a=a/b = 2
%=	a=a%b	a%=b; a=a%b = 0
>>=	a=a>>b	a=00000100 >> 2 = 00010000
<<=	a=a<<b	A=00000100 << 2 = 00000001
&=	a=a&b	(a=0100, b=0010) a&=b; a=a&b = 0000
|=	a=a|b	(a=0100, b=0010) a|=b; a=a|b =0110
^=	a=a^b	(a=0100, b=0010) a^=b; a=a^b = 0110

```
//Assignment  operator(+=,-=,*=,/=,%=,>>=,<<=,&=,|=,^=)
#include<stdio.h>
int main()
{
        int a,b;
        printf("Enter values:");
        scanf("%d%d",&a,&b);
        printf("\n a+=b  value is :%d",a+=b);
        printf("\n a-=b  value is :%d",a-=b);
        printf("\n a*=b  value is :%d",a*=b);
        printf("\n a/=b  value is :%d",a/=b);
        printf("\n a %=b  value is :%d",a%=b);
        printf("\n a<<=b  value is :%d",a<<=b);
        printf("\n a>>=b  value is :%d",a>>=b);

}
```

Output:

```
Enter values:100 2

 a+=b  value is :102
 a-=b  value is :100
 a*=b  value is :200
 a/=b  value is :100
 a =b  value is :0
 a<<=b  value is :0
 a>>=b  value is :0
```

```
//Assignment  operator(+=,-=,*=,/=,%=,>>=,<<=,&=,|=,^=)
#include<stdio.h>
int main()
{
        int a,b;
        printf("Enter values:");
        scanf("%d%d",&a,&b);
        printf("\n a<<=b  value is :%d",a<<=b);
        printf("\n a>>=b  value is :%d",a>>=b);

}
```

Output:

```
Enter values:100 2

 a<<=b  value is :400
 a>>=b  value is :100
```

```
//Assignment  operator(+=,-=,*=,/=,%=,>>=,<<=,&=,|=,^=)
#include<stdio.h>
int main()
{
        int a,b;
        printf("Enter values:");
        scanf("%d%d",&a,&b);
        //printf("a&=b value is :%d",a&=b);
        //printf("a|=b value is :%d",a|=b);
        printf("a^=b value is :%d",a^=b);
}
```

Output:

```
Enter values:5 3
a^=b value is :6
```

6.bitwise operators(&,|, ^,~,<<,>>)

Operator	Description	Example(a=1 and b=0)
&	bitwise AND	a & b = 0
|	bitwise OR	a| b = 1
^	bitwise XOR	a ^ b = 1
~	bitwise one's complement	~a = 0, ~b=1
<<	bitwise left shift, indicates the bits are to be shifted to the left.	1101 << 1 = 1010
>>	bitwise right shift, indicates the bits are to be shifted to the right.	1101 >> 1 = 0110

```
// bitwise operators(&,|, ^,~,<<,>>)
```

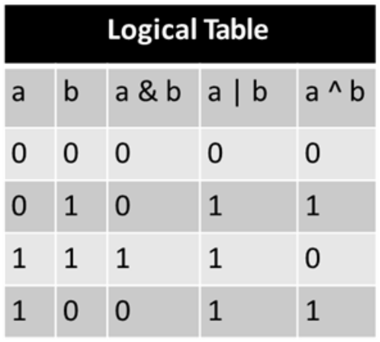
```
#include <stdio.h>
int main()
{
    int a = 5, b = 3;   // 5=101   3=011

    printf("a & b value is: %d\n", a & b);
    printf("a | b value is: %d\n", a | b);
    printf("a ^ b value is: %d\n", a ^ b);
    printf("~a value is: %d\n", ~a);
    printf("a >> b value is: %d\n", a >> b);
    printf("a << b value is: %d\n", a << b);

    return 0;
}
```

Output:

```
a & b value is: 1
a | b value is: 7
a ^ b value is: 6
~a value is: -6
a >> b value is: 0
a << b value is: 40
```



Logical Table

a	b	a & b	a | b	a ^ b
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

7.Conditional operator or Ternary operator(? :)

Conditional operator contains condition followed by two statements. If the condition is true the first statement is executed otherwise the second statement.

It is also called as ternary operator because it requires three operands.

Operator	Description	Example
?:	**conditional expression, Condition? Expression1: Expression2**	**(a>b)? “a is greater”: “b is greater”**

// Conditional operator or   Ternary operator(? :)

```
#include <stdio.h>
int main() {
  int age;


  printf("Enter your age: ");
  scanf("%d", &age);
  (age >= 18) ? printf("You can vote") : printf("You cannot vote");

  return 0;
}
```

Output:

```
Enter your age: 25
You can vote
```

8.Special operators(, sizeof, type)

Operator	Description	Example
,	comma operator, can be used to link the related expressions together	int a, b, x;
sizeof ()	sizeof operator to find the size of an object.	int a; sizeof(a)=2
type	Cast operator, to change the data type of the variable	float x= 12.5; int a; a = (int) x; value of a is 12.

```
// Special operators(, sizeof, type)
#include <stdio.h>

int main()
{
    int num = 10;
    printf("sizeof(num) = %d bytes\n",sizeof(num));
    printf("sizeof(10) = %d bytes\n", sizeof(10));
    printf("sizeof(int) = %d bytes\n",sizeof(int));
    printf("(float)num = %f\n", (float)num); //type operator
    return 0;
}
```

Output:

```
sizeof(num) = 4 bytes
sizeof(10) = 4 bytes
sizeof(int) = 4 bytes
(float)num = 10.000000
```