OPERATORS IN C WITH EXAMPLES

An OPERATOR is a symbol that tells the compiler to perform specific mathematical or logical functions. c language is built-in operators and provides to many operators are there:-

**TYPES OF C OPERATORS:**

C language offers many types of operators. They are,

1. Arithmetic operators
2. Assignment operators
3. Relational operators
4. Logical operators
5. Bitwise operators
6. Conditional operators (ternary operators)
7. Increment/decrement operators
8. Special operators

**CONTINUE ON TYPES OF C OPERATORS:**

Click on each operator name below for detailed description and example programs.

|  |  |
| --- | --- |
| **Types of Operators** | **Description** |
| [Arithmetic operators](http://fresh2refresh.com/c/c-operators-expressions/c-arithmetic-operators/) | These are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus |
| [Assignment operators](http://fresh2refresh.com/c/c-operators-expressions/c-assignment-operators/) | These are used to assign the values for the variables in C programs. |
| [Relational operators](http://fresh2refresh.com/c/c-operators-expressions/c-relational-operators/) | These operators are used to compare the value of two variables. |
| [Logical operators](http://fresh2refresh.com/c/c-operators-expressions/c-logical-operators/) | These operators are used to perform logical operations on the given two variables. |
| [Bit wise operators](http://fresh2refresh.com/c/c-operators-expressions/c-bit-wise-operators/) | These operators are used to perform bit operations on given two variables. |
| [Conditional (ternary) operators](http://fresh2refresh.com/c/c-operators-expressions/c-conditional-operators/) | Conditional operators return one value if condition is true and returns another value is condition is false. |
| [Increment/decrement operators](http://fresh2refresh.com/c/c-operators-expressions/c-increment-decrement-operators/) | These operators are used to either increase or decrease the value of the variable by one. |
| [Special operators](http://fresh2refresh.com/c/c-operators-expressions/c-special-operators/) | &, \*, size of ( ) and ternary operators. |

#### ****ARITHMETIC OPERATORS IN C:****

C Arithmetic operators are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus in C programs.

|  |  |
| --- | --- |
| **Arithmetic Operators/Operation** | **Example** |
| + (Addition) | A+B |
| – (Subtraction) | A-B |
| \* (multiplication) | A\*B |
| / (Division) | A/B |
| % (Modulus) | A%B |

#### ****EXAMPLE PROGRAM FOR C ARITHMETIC OPERATORS:****

In this example program, two values “40” and “20” are used to perform arithmetic operations such as addition, subtraction, multiplication, division, modulus and output is displayed for each operation.

#include <stdio.h>

Void main()

{

int a=40,b=20, add, sub,mul,div,mod;

add = a+b;

sub = a-b;

mul = a\*b;

div = a/b;

mod = a%b;

Printf ("Addition of a, b is : %d\n", add);

Printf ("Subtraction of a, b is : %d\n", sub);

printf ("Multiplication of a, b is : %d\n", mul);

printf("Division of a, b is : %

d\n", div);

printf("Modulus of a, b is : %d\n", mod);

}

#### ****OUTPUT:****

|  |
| --- |
| Addition of a, b is : 60 Subtraction of a, b is : 20 Multiplication of a, b is : 800 Division of a, b is : 2 Modulus of a, b is : 0 |

**ASSIGNMENT OPERATORS IN C:**

* In C programs, values for the variables are assigned using assignment operators.
* For example, if the value “10” is to be assigned for the variable “sum”, it can be assigned as “sum = 10;”

**EXAMPLE PROGRAM FOR C ASSIGNMENT OPERATORS:**

* In this program, values from 0 – 9 are summed up and total “45” is displayed as output.
* Assignment operators such as “=” and “+=” are used in this program to assign the values and to sum up the values.

|  |  |
| --- | --- |
| # include <stdio.h>    void main()  {  int a=10; //simple assignment  a+=10; //compound ass. (a=a+10)  a-=10;  a%=10; etc….  }   |  | | --- | |  |   printf(“%d”, a);  getch();  } |

#### ****OUTPUT:****

here dffrent 2 ans. will be appears

**RELATIONAL OPERATORS IN C:**

Relational operators are used to find the relation between two variables. i.e. to compare the values of two variables in a C program.

|  |  |
| --- | --- |
| **Operators** | **Example/Description** |
| > | x > y (x is greater than y) |
| < | x < y (x is less than y) |
| >= | x >= y (x is greater than or equal to y) |
| <= | x <= y (x is less than or equal to y) |
| == | x == y (x is equal to y) |
| != | x != y (x is not equal to y) |

**EXAMPLE PROGRAM FOR RELATIONAL OPERATORS IN C:**

* In this program, relational operator (==) is used to compare 2 values whether they are equal are not.
* If both values are equal, output is displayed as ” values are equal”. Else, output is displayed as “values are not equal”.
* Note : double equal sign (==) should be used to compare 2 values. We should not single equal sign (=).

#### #include<stdio.h>

#### #include<conio.h>

#### void main()

#### {

#### int a=10,b=20;

#### clrscr();

#### if(a==b) if(a!=b) if(a%=b) etc.

#### {

#### printf(“both r equals”);

#### }

#### else

#### {

#### printf(“both r not equals”)

#### }

#### grtch();

#### }

**LOGICAL OPERATORS IN C:**

* These operators are used to perform logical operations on the given expressions.
* There are 3 logical operators in C language. They are, logical AND (&&), logical OR (||) and logical NOT (!).

|  |  |
| --- | --- |
| **Operators** | **Example/Description** |
| && (logical AND) | (x>5)&&(y<5) It returns true when both conditions are true |
| || (logical OR) | (x>=10)||(y>=10)  It returns true when at-least one of the condition is true |
| ! (logical NOT) | !((x>5)&&(y<5))  It reverses the state of the operand “((x>5) && (y<5))”  If “((x>5) && (y<5))” is true, logical NOT operator makes it false |

|  |
| --- |
|  |

#include <stdio.h>

#include<conio.h>

void main()

{

int a=10,b=20;

clrscr();

//&& (a&&b) (logical and operator) (a||b) ||(logical or operator) !(

**1 .printf(“%d”,a&&b); //show true value if both values are true print 1 for true…….like**

**printf(“%d”,(a<20&&b>10));**

**2 printf(“%d”,(a>20||b<10)); // at least one value have to true then show true value……**

**3 printf(“%d”,!(a<20)&&(b>10)); !(always return reverse value like true ka false or false aya to true)**

**printf(“%d”,!(a<20)||(b>10));**

**CONDITIONAL OR TERNARY OPERATORS IN C:**

* Conditional operators return one value if condition is true and returns another value is condition is false.
* This operator is also called as ternary operator.

ternary operators(conditional operators)

#include<stdio.h>

#include<conio.h>

void main()

{

//clrscr();

int a,b;

clrscr();

printf(“enter two numbers”);

scanf(“%d%d”, &a,&b);

(a>=b)?printf(“%d is >= %d\n”,a,b):printf(“%d is < %d\n”,a,b);

getch();

}

output will be show greatest value of integer through ternary or conditional (operator).

INCREMENT & DECREMENT OPERATORSIN C :

/\* post increment a++

/\*pre increment ++a

/\* post decrement a--

/\* pre decrement --a

#include<stdio.h>

Main()

{

Int a=5;

Printf(“post increment %d” , a++); //5 after increase a=6

Printf(“pre increment %d” , ++a); //7 pre increase than a=7

#### Printf(“post decrement %d” , a--); // 7 after decrease then a=6

#### Printf(“post increment %d” , --a); //5 pre dec then value a=5

#### ****so output will be****

#### ****post increment: 5****

#### ****pre increment: 7****

#### ****post decrement: 7****

#### ****pre decrement: 5****

#### ****SPECIAL OPERATORS IN C:****

Below are some of the special operators that the C programming language offers.

|  |  |
| --- | --- |
| **Operators** |  |
| Sizeof () | This gives the size of the variable.  Example : size of (char) will give us 1. |
| (comma), | , operator |
|  |  |

* 1. #include<stdio.h>

Main()

{

Int x;

Printf(“memory status: %d”,sizeof(x));

}

Output will be show parricular size about any data like int store 2 bytes in memory like that.

2 comma operator examples..

If we want to write very short formate of any programming structure than we can easily use comma operator in c ..

* 1. #include<stdio.h>

Main()

{

Int a=2,b=3,c=4,d=a+b+c;

Printf(“result is %d”,d);

}

Output will be d=9

Bitwise operators

which are bacically performed bit level operations task.

performed bit to bit translation of any interger value.

there are 5 types

* 1. AND BITWISE OPERATOR (&)
* truth table for and bitwise operator

|  |  |  |
| --- | --- | --- |
| operands | operands | Result |
| true(1) | true | true |
| true | false | false(0) |
| false | true | false |
| false | false | false |

this task firstly convert decimal to binary then convert them according to own truth table format. like here an example.

program

#include<stdio.h>

void main()

{

int a=60; // 00111100 (binary of 60) (compare both binary format according to truth table of bitwise and& operator then jo bhi binary formate aayga usko decimal me conver krke he console pr output ,milega.)

int b=17; // 00010001 (binary of 17)

printf(“result of bitwise AND operator: %d”,a&b);

}

so out put will be : 16 hoga

* 1. BITWISE OR(|) OPERATOR:-
* truth table for and bitwise operator

|  |  |  |
| --- | --- | --- |
| operands | operands | Result |
| true(1) | true | true |
| true | false | true |
| false | true | true |
| false | false | false |

this task firstly convert decimal to binary then convert them according to own truth table format. like here an example.

program

#include<stdio.h>

void main()

{

int a=60; // 00111100 (binary of 60) (compare both binary format according to truth table of bitwise and& operator then jo bhi binary formate aayga usko decimal me conver krke he console pr output ,milega.)

int b=17; // 00010001 (binary of 17)

printf(“result of bitwise or operator: %d”,a|b);

}

* 1. BITWISE XOR(^) OPERATOR:-
* truth table for and bitwise operator

|  |  |  |
| --- | --- | --- |
| operands | operands | Result |
| true(1) | true | false |
| true | false | true |
| false | true | true |
| false | false | false |

this task firstly convert decimal to binary then convert them according to own truth table format. like here an example.

program

#include<stdio.h>

void main()

{

int a=60; // 00111100 (binary of 60) (compare both binary format according to truth table of bitwise and& operator then jo bhi binary formate aayga usko decimal me conver krke he console pr output ,milega.)

int b=17; // 00010001 (binary of 17)

printf(“result of bitwise Xor operator: %d”,a^b);

}

output will be about decimal format about truth table recognization.

BITWISE INVERSION OOPERATOR( ONES COMPLEMENT ) ~

its unique type of bit level operator . ye de gai value me 1 byte add krke usko minus me show krwayga jese int a=5 he to output will be -6 hoga…

program

#include<stdio.h>

void main()

{

int a=5;

printf(“RESULT OF INVERSION IS %d”, ~a);

}

HERE output will be -6

BITWISE LEFT SHIFT << OPERATOR

(left shift the bits to left side)

program

#include<stdio.h>

main()

{

int a=60; // 00111100 (60 ka binary) ka 2 bite shift hoga left side )

to result hoga 11110000 iska DECIMAL HOGA 240

printf(“RESULT IS %d”, a<<2);

}

output will be 240 (11110000 (BINARY ME) YE MEANING HOGA 240 KA.

BITWISE RIGHT SHIFT >> OPERATOR

(RIGHT shift the bits to left side)

program

#include<stdio.h>

main()

{

int a=60; // 00111100 (60 ka binary) ka 2 bite shift hoga left side )

to result hoga 11110000 iska DECIMAL HOGA 240

printf(“RESULT IS %d”, a>>2);

}

output will be 240 (11110000 (BINARY ME) YE MEANING HOGA 240 KA.

**DECISION MAKING AND BRANCHINGS IN C**

HERE SO MANY TYPES OF DECISION MAKING BRABCHES AND STATEMENTS…

* 1. IF STATEMENT
  2. SWITCH STATEMENT
  3. GOTO STMT
  4. CONTINUE STMT
  5. BREAK STMT
  6. **IF STATEMENT**

IF only one conditional programs we can easily use if stmt….

Syntax: IF(CONDITION)

{ ENTRY

STMT TRUE

CONDITION

}

STMT

WAP TO FINF EVEN NUMBER PROGRAM. FALSE

STMT

#include<stdio.h>

main()

NEXT STMT

{

int a;

clrscr();

printf(“enter a no”);

scanf(“%d”,&a);

if(a%2==0)

{

printf(“even no %d”,a);

}

printf(“value of A %d”,a);

getch();

}

opuput will be define only even number…

* 1. **IF else STATEMENT**

Its used to decision making. It executes a statement based on result of any double expression.

IF(condition.)

stmt 1;

else

stmt 2

if  the if exp. Executed to true, then stmt 1 or block will be executes,otherwise stmt 2 (else)will be executed.

WAP TO find greatest value

WAP TO find even or odd numbers.

#include<stdio.h>

main()

{

int a;

clrscr();

printf(“enter a no”); /\*input printf

scanf(“%d”,&a); / 0 name k index me a variable store ho gya he.

if(a%2==0)

{

printf(“even no %d”,a); //output printf

}

else

{

printf(“number is odd”,a);

}

}

2. FIND GREATEST VALUE

#include<stdio.h>

Main()

{

Int x=3,y=4;

If(x>y)

{

Printf(“x is greater”);

}

Else

{

Printf(“y is greater”);

}

}

* 1. **NESTED IF ELSE STATEMENT**

First if block used 2nd if block then will be can say nested if statement started….

IF(COND. 1) // OUTER IF STMT

{

IF(COND.2)

SUB IF PART MEANS NESTED PART

{

BLOCK 1;

}

ELSE

{

BLOCK 2;

}

}

ELSE

FINAL ELSE PART(AGER OUTER IF FALUSE HOGA TO SIDHA YE ELSE PART RUN HOGA

{

BLOCK 3;

}

Firstly if a ko compare karte he b se . a ki value badi hue to true cod. Hone se loop 2 sub if me jayga means nested part me. Or con. 2 checkn krwai jaygi.otherwise if outer he false hota he to siddha skip hokar main else part pr jump karenge.

WAP TO FIND GREATEST VALUES. Of A,b,c

#include<stdio.h>

Main()

{

Int a,b,c;

Printf(“enter three values”);

Scanf(“%d%d%d”,&a,&b,&c);

// comparision strat

If(a>b)

{

If(a>c)

{

Printf(“greater value %d”,a);

}

}

Else

{

If(b>c)

{

Printf(“greater value %d”b);

}

Else

{

Printf(“greatest value%d”,c);

}

}

}

WAP TO FIND LEAP YEAR….USING NESTED IF ELSE

2000

year

y%4

2000%4

// Remender will be 0

Remender will be 0

2000%100

Y%100

2000%100

Y%400

Remender will be 0

2012 k remender me condition 2012%4 me true hogi remender 0 hoga but niche ki process me 2012%100 remender 0 na hone k wajah se program baher hoga but year leap he hoga.. because loop nested h. accoding to conditions.

c

int y;

printf(“enter a year”);

scanf(“%d”,&y);

if(y%4==0)

{

if(y%100==0)

{

if(y%400==0)

{

printf(“%d is leap year”,y);

}

else

{

printf(“%d is not leap year”,y);

}

}

else

{

printf(“%d is leap year”,y);

}

else

{

printf(“%d is not leap year”,y);

}

getch();

}

if we can input 3 year 2000 (leap year), 2012(leap year),1900(not a leap year) so here three types of process(y%4==0), (y%100==0), (y%400==0)

4**.ELSE IF LADDER**

We have 4 codditions then we have to use 4 block also. Jis condition se jo block match ho jaye wo he block execute….

Syntax:-

IF(COND.1)

{

BLOCK 1;

}

ELSE IF(COND 2)

{

BLOCK 2;

}

ELSE IF(CON.3)

{

BLOCK3;

}

WAP TO FIND GRADE USING MARKS :-

#include<stdio.h>

Main()

{

Int marks;

Printf(“enter the marks”);

Scanf(“%d”,&marks);

If(marks<=34)

Printf(“grade F”);

Else if(marks<=45)

Printf(“grade E”);

Else if(marks<=59)

Printf(“grade D”);

Else if(marks<=69)

Printf(“grade C”);

Else if(marks<=79)

Printf(“grade B”);

Else if(marks<=89)

Printf(“grade A”);

Else

Printf(“grade s”);

}

WAP to choose any 3 value and after 3 have to exit by default.

#include<stdio.h>

Main()

{

Int ch;

Clrscr();

Printf(“\n ADD”);

Printf(“\n ADD”);

Printf(“\n ADD”);

Printf(“\n ADD”);

Printf(“\n enter your choice”);

Scanf(“%d”,&ch);

If(ch==1)

{

Printf(“add option selected”);

}

Else if(ch==2)

{

Printf(“edit option selected”);

}

Else if(ch==3)

{

Printf(“delete option selected”);

}

Else if(ch==4)

{

Printf(“exit option selected”);

}

Else

{

Printf(“invalid entry”);

}

Getch();

}

**SWITCH STATEMENT:-**

It’s a multiway decision making that tests the value of any exp. When match is found the stmt associated with that are executed.

Syntax: switch(expression)

{

Case value 1:

Block 1;

Break;

Case value 2:

Block 2;

Break;

Default:

Default block;

Break;

}

WAP TO PRINT VOWELS OR CONSONAMNT USING SWITCH CASE.

#include<stdio.h>

#include<conio.h>

Void main()

{

Char ch;

Clrscr();

Printf(“enter a character”);

Scanf(“%c”,&ch);

Switch(ch)

{

Case ‘a’:

Case ‘e’:

Case ‘i’:

Case ‘o’:

Case ‘u’:

Printf(“\n %c is a vowels”,ch);

Break;

Default:

Printf(“%c is consonant”,ch);

Break;

}

Getch();

}

.WAP TO CREATE MENU DRIVEN PROGRAM (switch case)

#include<stdio.h>

#include<conio.h>

Void main()

{

int choice,a,b,s;

while(1) //non(0) constant evaluate ho to while loop k througt in infinite loop start hoga) prog..

{

Clrscr();

printf

Printf(“\n1). ADDITION”);

Printf(“\n2). Odd-even”);

Printf(“\n3).printing first no”);

Printf(“/n4) Exit”); //exit button banaker console se program baar baar code par jump nahi hoga.

Printf(“enter your choice”);

Scanf(“%d”,&choice);

Switch(choice)

{

Case 1:

Printf(“Enter two numbers”);

Scanf(“%d%d”,&a,&b);

S=a+b;

Printf(“sum is %d\n”,s);

Break;

Case 2:

Printf(“Entera numbers”);

Scanf(“%d”,&a,);

If(a%2==0)

{

Printf(“even no”);

}

Else

Printf(“odd no”);

Break;

Case 3:

Printf(“Entera numbers”);

Scanf(“%d”,&a,);

For(b=1;b<=a;b++)

Printf(“%d”,b);

Break;

Case 4: exit(0); //optional he kyki jab hum while ka use karte he to pre define function hota h. jisse hum console se code par jump nahi hote..

Default:

Printf(“invalid choice”);

}

Getch();

}

**HOW TO LEARN GO TO, CONTINUE, BREAK in C.**

**CONTINUE**

#include<stdio.h>

#include<conio.h>

Void main()

{

Int i;

Clrscr();

For(i=1;i<=10;i++)

{

If(i==3) // we want to skip particular numbers.

{

Continue; //

}

Printf(“%d\n”,i);

}

Getch();

}

**break**

#include<stdio.h>

#include<conio.h>

Void main()

{

Int i;

Clrscr();

For(i=1;i<=10;i++)

{

If(i==5)

Break;

}

Printf(“exited from the for loop”);

Getch();

}

**GOTO**

#include<stdio.h>

#include<conio.h>

Void main()

{

Clrscr();

Printf(“\n hello”);

{

Goto g; // g is a label name of goto statement

}

Printf(“\n plz don’t skip me”);

G: // colon sign with label

Printf(“\n yes u r skipeddddd”);

Getch();

}

Output will be : yes u r skipedddddd.

**LOOPs IN C :-**

Loops follows the iteration concepts.looping allows a setoff instructions to be performed until a certain condition becomes false.

1. FOR LOOP
2. WHILE LOOP
3. DO-WHILE LOOP

1.FOR LOOP:- it’s a entry controlled loop that provides that kinds of control structure.

For(initialization; test condition ;inc/dec)

**{**

STATEMENT;

}

WAP TO PRINT 1 TO 100 VALUES using for loop

Or

Wap to find prime number or not using for loop

#include<stdio.h>

#include<conio.h>

Main()

{

Int n,i;

Input- 13 input 9

Computations:

I=2,3,4,5,6

13 13 9 9

Printf(“enter a no.”);

Scanf(“%d”,&n);

For(i=2,i<=n/2;i++)

{

If(n%i==0)

{

Printf(“%d is non-prime”,n);

Exit(0); //optional.

3 

2 

3 

2

}

}

9 is a non prime no

13 cant devide so its prime no

Printf(“%d is prime”,n);

}

WAP to print that numerical series…using for loop.

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

#include<stdio.h>

Main()

{

Int I,j;

For(i=1;i<=5;i++)

{

For(j=1;j<=I;j++)

{

Printf(“%d”,i);

}

Printf(“\n”);

}

}

WAP TO PRINT EACH ELEMENTS IN METRIX USING FOR LOOP.

#include<stdio.h>

Main()

{

Int n,m,i,j;

Printf(“enter the value of m and n\n”);

Scanf(“%d%d”,&m,&n);

Printf(“to print matrix locations\n”);

For(i=0;i<m;i++)

{

For(j=0;j<n;j++)

{

Printf(“%d%d”,I,j);

}

Printf(“\n”);

}

}

* 1. **WHILE LOOP.**

A set of statement may have to be repreatedly executed for a specific numbers of times,in such situation, we use looping area.

Here difference b/w while &do- while loop.

Do

{

Stmt 1;

Stmt 2;

Stmt n;

}

While(exp)

While loop

While(exp)

{

Stmt 1;

Stmt 2;

Stmt n;

}

For example:-

While do-while

i=0; i=0;

sum=0; sum=0;

while(i<=n) do

{

Sum=sum+1;

{ i=i+1;

Sum=sum+I; }

i= i+1; while(i<=n);

}

**EXAMPLES WHILE LOOP:-**

WAP TO PRINT INTEGER THROUGH 0 TO 9;

#include<stdio.h>

{

Int digit=0;

While(digit<=9)

{

Printf(“%d\n”,digit);

++digit;

}

}

WAP TO REPRESENT ARMSTRONG NUMBERS THROUGT INPUT.

#include<stdio.h>

#include<conio.h>

Void main()

{

Int input,sum=0,temp;

Printf(“plz enter a no”);

Scanf(“%d”,&input);

Temp=input;

While(temp!=0)

{

Int r=temp%10;

Sum=ssum+r\*r\*r;

Temp=temp/10;

}

If(sum==input)

Printf(“%d is an amstrong no,”,input);

Else

Printf(“%d is not an amstrong no”,input);

}

Getch();

**Do-while loop example :**

WAP TO FIND SUM OF SERIES USING DO WHILE LOOP

#include<stdio.h>

Main()

{

Int n,I,sum;

Printf(“enter the value of n\n”);

Scanf(“%d”,&n);

Sum=i=0;

Do

{

Sum=sum+i;

I++;

}

While(i<=n);

Printf(“Sum of series=%d”,sum);

}

WAP TO SUM OF GIVEN NO WITH REVERSE NUMBERS

#include<stdio.h>

Main()

{

Int n,digit,sum,rev;

Printf(“Enter positive integer number\n”);

Scanf(“%d”,&n);

Sum=rev=0;

Do

{

Digit=n%10;

Sum+=digit;

Rev=rev\*10+digit;

N=n/10;

}

While(n!=0);

Printf(“sum of digits=%d\n”,sum);

Printf(“reverse numbers=%d\n”,rev);

}

Output will be

If We have enterd 523 sum of digits is 10 & reverse no= 325.