

Eg1: $6 * 2 / (2 + 1 * 2 / 3 + 6) + 8 * (8 / 4)$ ----- 17

$$6 * 2 / (2 + 2 / 3 + 6) + 8 * (8 / 4)$$

$$6 * 2 / (2 + 0 + 6) + 8 * (8 / 4)$$

$$6 * 2 / (2 + 6) + 8 * (8 / 4)$$

$$6 * 2 / 8 + 8 * (8 / 4)$$

$$6 * 2 / 8 + 8 * 2$$

$$12 / 8 + 8 * 2$$

$$1 + 8 * 2$$

$$1 + 16$$

$$17$$

Eg2: $a + b * a / b - a \% b$ ----- $a=10, b=2$ ----- 20

$$10 + 2 * 10 / 2 - 10 \% 2$$

$$10 + 20 / 2 - 10 \% 2$$

$$10 + 10 - 10 \% 2$$

$$10 + 10 - 0$$

$$20 - 0$$

Eg3: $10 - 3 \% 8 + 6 / 4$ ----- 8

$$10 - 3 \% 8 + 6 / 4$$

$$10 - 3 + 6 / 4$$

$$10 - 3 + 1$$

$$7 + 1$$

$$8$$

Eg4: $2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8$ -----8

Example 1: Determine the hierarchy of operations and evaluate the following expression:

$$i = 2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8$$

Stepwise evaluation of this expression is shown below:

$$i = 2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8$$

$$i = 6 / 4 + 4 / 4 + 8 - 2 + 5 / 8$$

operation: *

$$i = 1 + 4 / 4 + 8 - 2 + 5 / 8$$

operation: /

$$i = 1 + 1 + 8 - 2 + 5 / 8$$

operation: /

$$i = 1 + 1 + 8 - 2 + 0$$

operation: /

$$i = 2 + 8 - 2 + 0$$

operation: +

$$i = 10 - 2 + 0$$

operation: +

$$i = 8 + 0$$

operation: -

$$i = 8$$

operation: +

Eg5: $17 - 8 / 4 * 2 + 3 - ++a$ -----a=5-----10

$$17 - 8 / 4 * 2 + 3 - ++a$$

$$17 - 8 / 4 * 2 + 3 - 6$$

$$17 - 2 * 2 + 3 - 6$$

$$17 - 4 + 3 - 6$$

$$13 + 4 - 6$$

$$16 - 6$$

$$10$$

Eg6: $2 * ((a \% 5) * (4 + (b - 3) / (c + 2)))$ ---- $a=8, b=15, c=4$ ---- 36

$$2 * ((a \% 5) * (4 + (b - 3) / (c + 2)))$$

Evaluate the Expression by assuming $a=8$ $b=15$ $c=4$

Solution

$$\begin{aligned} & 2 * ((a \% 5) * (4 + (b - 3) / (c + 2))) \\ \rightarrow & 2 * ((8 \% 5) * (4 + (15 - 3) / (4 + 2))) \\ & 2 * (3 * (4 + (15 - 3) / (4 + 2))) \\ & 2 * (3 * (4 + 12 / (4 + 2))) \\ & 2 * (3 * (4 + 12 / 6)) \\ & 2 * (3 * (4 + 2)) \\ & 2 * (3 * 6) \\ & 2 * 18 \Rightarrow \boxed{36} \end{aligned}$$

Eg7: $100 / 20 <= 10 - 5 + 100 \% 10 - 20 == 5 >= 1 != 20$ ----- 1

$$\begin{aligned} & \boxed{100 / 20} <= 10 - 5 + 100 \% 10 - 20 == 5 >= 1 != 20 \\ & 5 <= 10 - 5 + \boxed{100 \% 10} - 20 == 5 >= 1 != 20 \\ \rightarrow & 5 <= \boxed{10 - 5} + 0 - 20 == 5 >= 1 != 20 \\ & 5 <= \boxed{5 + 0} - 20 == 5 >= 1 != 20 \\ & 5 <= \boxed{5 - 20} == 5 >= 1 != 20 \\ & \boxed{5 <= -15} == 5 >= 1 != 20 \\ & 0 == \boxed{5 >= 1} != 20 \\ & 0 == \boxed{1} != 20 \\ & 0 != 20 \Rightarrow \boxed{1} \end{aligned}$$

Eg8: $a+2>b\&\&!c||a!=d\&\&a-2<=e$, $a=11, b=6, c=0, d=7, e=5$ ----1

$$\begin{aligned}
 &100/20 <= 10-5 + 100\%10-20 == 5 >= 1 != 20 \\
 &5 <= 10-5 + 100\%10-20 == 5 >= 1 != 20 \\
 &\rightarrow 5 <= 10-5 + 0-20 == 5 >= 1 != 20 \\
 &5 <= 5+0-20 == 5 >= 1 != 20 \\
 &5 <= 5-20 == 5 >= 1 != 20 \\
 &5 <= -15 == 5 >= 1 != 20 \\
 &0 == 5 >= 1 != 20 \\
 &0 == 1 != 20 \\
 &0 != 20 \Rightarrow 1
 \end{aligned}$$

Operator	Priority	Associativity
{}, (), []	1	Left to right
++, --, !	2	Right to left
*, /, %	3	Left to right
+, -	4	Left to right
<, <=, >, >=, ==, !=	5	Left to right
&&	6	Left to right
	7	Left to right
?:	8	Right to left
=, +=, -=, *=, /=, %=	9	Right to left

Pre increment and post increment

```
p = 1;
int q = p++; // q = 1
int r = p; // r = 2
```

`%[flag][min width][precision][length modifier][conversion specifier]`

`-,0—flags`

`.precision(Max) and width (min)`

```
printf( "%.3f", 1.2 );--1.200
```

```
printf( "%.3f", 1.2348 );--1.235
```

```
printf( "%8.5f\n", 1.234 );--1.23400
```

Eg: program to arithmetic operators

```
// Working of arithmetic operators
#include <stdio.h>
int main()
{
    int a = 9, b = 4, c;

    c = a+b;
    printf("a+b = %d \n",c);
    c = a-b;
    printf("a-b = %d \n",c);
    c = a*b;
    printf("a*b = %d \n",c);
    c = a/b;
    printf("a/b = %d \n",c);
    c = a%b;
    printf("Remainder when a divided by b = %d \n",c);

    return 0;
}
```

Increment and Decrement operators

```
#include <stdio.h>
int main()
{
    int a = 10, b = 100;
    float c = 10.5, d = 100.5;

    printf("++a = %d \n", ++a);
    printf("--b = %d \n", --b);
    printf("++c = %f \n", ++c);
}
```

```

printf("--d = %f \n", --d);

return 0;
}

```

Program for assignment operators

```

// Working of assignment operators
#include <stdio.h>
int main()
{
    int a = 5, c;

    c = a;      // c is 5
    printf("c = %d\n", c);
    c += a;     // c is 10
    printf("c = %d\n", c);
    c -= a;     // c is 5
    printf("c = %d\n", c);
    c *= a;     // c is 25
    printf("c = %d\n", c);
    c /= a;     // c is 5
    printf("c = %d\n", c);
    c %= a;     // c = 0
    printf("c = %d\n", c);

    return 0;
}

```

Program for relational operators

```

// Working of relational operators
#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10;

    printf("%d == %d is %d \n", a, b, a == b);
    printf("%d == %d is %d \n", a, c, a == c);
    printf("%d > %d is %d \n", a, b, a > b);
    printf("%d > %d is %d \n", a, c, a > c);
    printf("%d < %d is %d \n", a, b, a < b);
    printf("%d < %d is %d \n", a, c, a < c);
    printf("%d != %d is %d \n", a, b, a != b);
    printf("%d != %d is %d \n", a, c, a != c);
    printf("%d >= %d is %d \n", a, b, a >= b);
    printf("%d >= %d is %d \n", a, c, a >= c);
    printf("%d <= %d is %d \n", a, b, a <= b);
    printf("%d <= %d is %d \n", a, c, a <= c);
}

```

```
    return 0;
}
```

Program for logical operators

```
// Working of logical operators

#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10, result;

    result = (a == b) && (c > b);
    printf("(a == b) && (c > b) is %d \n", result);

    result = (a == b) && (c < b);
    printf("(a == b) && (c < b) is %d \n", result);

    result = (a == b) || (c < b);
    printf("(a == b) || (c < b) is %d \n", result);

    result = (a != b) || (c < b);
    printf("(a != b) || (c < b) is %d \n", result);

    result = !(a != b);
    printf("!(a != b) is %d \n", result);

    result = !(a == b);
    printf("!(a == b) is %d \n", result);

    return 0;
}
```

Examples of Explicit Type Casting in C

```
#include<stdio.h>

int main()
{
    float num = 56.3;
    int p = (int)num + 50; // data type casting explicitly
    printf("Let us understand Explicit Type Casting in C\n");
    printf("The value of the digit used is: %f\n", num);
    printf("The value of the variable p is: %d\n",p);
    return 0;
}
```

```
}
```

Implicit type casting

```
#include<stdio.h>
int main(){
    short a=10; //initializing variable of short data type
    int b; //declaring int variable
    b=a; //implicit type casting
    printf("%d\n",a);
    printf("%d\n",b);
}
```

If

```
// Program to display a number if it is negative
```

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

    printf("Enter an integer: ");
    scanf("%d", &number);

    // true if number is less than 0
    if (number < 0) {
        printf("You entered %d.\n", number);
    }

    printf("The if statement is easy.");

    return 0;
}
```

If else

```
// Check whether an integer is odd or even
```

```
#include <stdio.h>
int main() {
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);

    // True if the remainder is 0
    if (number%2 == 0) {
```

```

        printf("%d is an even integer.",number);
    }
    else {
        printf("%d is an odd integer.",number);
    }

    return 0;
}

```

If else if else

```

// Program to relate two integers using =, > or < symbol

#include <stdio.h>
int main() {
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    //checks if the two integers are equal.
    if(number1 == number2) {
        printf("Result: %d = %d",number1,number2);
    }

    //checks if number1 is greater than number2.
    else if (number1 > number2) {
        printf("Result: %d > %d", number1, number2);
    }

    //checks if both test expressions are false
    else {
        printf("Result: %d < %d",number1, number2);
    }

    return 0;
}

```

Nested if

```

#include <stdio.h>
int main() {
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    if (number1 >= number2) {
        if (number1 == number2) {
            printf("Result: %d = %d",number1,number2);
        }
        else {

```

```

        printf("Result: %d > %d", number1, number2);
    }
}
else {
    printf("Result: %d < %d", number1, number2);
}

return 0;
}

```

Loops :

While

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i = 20;
    while( i <=20 ) {
        printf ("%d " , i );
        i++;
    }
    getch();
}

```

Do while

```

do{
    //code
}while(condition);

```

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i = 20;
    do{
        printf ("%d " , i );
        i++;
    }
    while( i <=20 );
    getch();
}

```

For loop

- Initialization
- Condition
- Increment/Decrement

```
• for(initialization;condition;increment/decrement)
• {
• //code
• }
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i;
for( i = 20; i < 25; i++) {
printf ("%d " , i);
}
getch();
}
```

```
#include <stdio.h>

int main() {

    double n1, n2, n3;

    printf("Enter three different numbers: ");
    scanf("%lf %lf %lf", &n1, &n2, &n3);

    // if n1 is greater than both n2 and n3, n1 is the largest
    if (n1 >= n2 && n1 >= n3)
        printf("%.2f is the largest number.", n1);

    // if n2 is greater than both n1 and n3, n2 is the largest
    if (n2 >= n1 && n2 >= n3)
        printf("%.2f is the largest number.", n2);

    // if n3 is greater than both n1 and n2, n3 is the largest
    if (n3 >= n1 && n3 >= n2)
        printf("%.2f is the largest number.", n3);

    return 0;
}
```

```

#include <stdio.h>

int main() {

    double n1, n2, n3;

    printf("Enter three numbers: ");
    scanf("%lf %lf %lf", &n1, &n2, &n3);

    // if n1 is greater than both n2 and n3, n1 is the largest
    if (n1 >= n2 && n1 >= n3)
        printf("%.2lf is the largest number.", n1);

    // if n2 is greater than both n1 and n3, n2 is the largest
    else if (n2 >= n1 && n2 >= n3)
        printf("%.2lf is the largest number.", n2);

    // if both above conditions are false, n3 is the largest
    else
        printf("%.2lf is the largest number.", n3);

    return 0;
}

```

```

#include <stdio.h>

int main() {

    double n1, n2, n3;

    printf("Enter three numbers: ");
    scanf("%lf %lf %lf", &n1, &n2, &n3);

    // outer if statement
    if (n1 >= n2) {

        // inner if...else
        if (n1 >= n3)
            printf("%.2lf is the largest number.", n1);
        else
            printf("%.2lf is the largest number.", n3);
    }

    // outer else statement
    else {

```

```
// inner if...else
if (n2 >= n3)
    printf("%.2lf is the largest number.", n2);
else
    printf("%.2lf is the largest number.", n3);
}

return 0;
}
```