Tutorial Link https://codequotient.com/tutorials/C - Relational and Logical Operators/5a1d98bb52795c1b16c0ac00

TUTORIAL

C - Relational and Logical Operators

Chapter

1. C - Relational and Logical Operators

Topics

1.1 Relational Operators

1.3 Logical operators

Relational Operators

In C language, we can define true and false outputs based on some conditions. Any value other than 0 (ZERO) in C is treated as true, whereas ZERO is false. Be careful, all positive and negative numbers other than 0 are true, only 0 is treated as false in C language.

Relational operators return 0 when they found a false result, and 1 if they found a true result based on their conditions. Following are the relational operators: -

```
Operators   Description
>           greater than
>=          greater than or equal
<           less than
<=          less than or equal
==          equal
!=          not equal
```

We can check any two expressions with these relational operators. For example,

```
4 > 5;      // return 0 as condition is false.
4 == 4;      // return 1 as they match
3 != 6;      // return 1 as condition is true.
4 <= 9;      // return 1 as condition satisfies
```

```c
#include <stdio.h>

int main()
{
  int a,b,c,d;
  a=5;
  b=2;
  c=5;
  d=8;

  printf("a=%d b=%d c=%d d=%d \n\n",a,b,c,d);
  if(a > b) printf("a is greater than b.\n");
  if(a >= d) printf("a is greater than or equal to d.\n");
  if(a == c) printf("a is equal to c.\n");
  if(a != d) printf("a is not equal to d.\n");
  if(d <= c) printf("d is less than or equal to c.\n");
  if(d > c) printf("d is greater than c.\n");

  return 0;
}

```

Logical operators

These are similar to relational operators and are used to combine the conditions. Following are the logical operators in C: -

```
Operators   Descriptions
&&           AND
||           OR
!            NOT
```

These operators behave in same manner as they do in boolean algebra. There truth table is below: -

```
A   B   A&&B  A||B   !A
0   0   0      0     1
0   1   0      1     1
1   0   0      1     0
1   1   1      1     0
```

In C, from values, operands will be evaluated to either 0 or 1. These are used to combine relational operations, for example,

```
(3 > 1) && (5 != 6)   // combining two conditions. Evaluated as (1
&& 1) = 1
(1 > 3) && (5 != 6)   // combining two conditions. Evaluated as (0
&& 1) = 0
(1 > 3) || (5 != 6)   // combining two conditions. Evaluated as (0
|| 1) = 1
```

```c
#include <stdio.h>

int main()
{
  int a,b,c,d;
  a=5;
  b=2;
```

```
  c=5;
  d=8;

  printf("a=%d b=%d c=%d d=%d \n\n",a,b,c,d);

  if((a > b) && (a == c)) printf("a is greater than b AND a is equal to c.\n");
  if((a <= b) && (d >= b)) printf("a is less than b AND d is greater than b.\n");
  if((a != b) && (a <= d)) printf("a is not equal to b AND a is less than or equal to d.\n");

  return 0;
}

```