
Subject: Logical operators

Posted by [Gompie](#) on Sun, 13 Nov 2016 14:30:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi

I am trying to do

```
if( ( a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50)) then begin
```

```
c=! NANQ
```

...but the multiplie logical operator not working.

Can you help

GlanPlon

Subject: Re: Logical operators

Posted by [Jim Pendleton](#) on Sun, 13 Nov 2016 15:40:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Sunday, November 13, 2016 at 7:30:50 AM UTC-7, Glan wrote:

> Hi

> I am trying to do

> if((a gt 50) and (b lt 50)) or ((a lt 50) and (b gt 50)) then begin

> c=! NANQ

>

> ...but the multiplie logical operator not working.

> Can you help

> GlanPlon

You don't say what "not working" means, but my first guess is that the variables a and b are not both scalars. You may need to re-cast your problem using a vector operation, a WHERE or a loop.

Jim P

Subject: Re: Logical operators

Posted by [Gompie](#) on Sun, 13 Nov 2016 16:18:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks

By not working, I mean the conditions are not applied and I get C=!NANQ everywhere.

Can logical operators be combined in idl or not.

Glan

On Sunday, November 13, 2016 at 10:40:46 AM UTC-5, Jim P wrote:

> On Sunday, November 13, 2016 at 7:30:50 AM UTC-7, Glan wrote:

>> Hi

```
>> I am trying to do
>> if( ( a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50)) then begin
>> c=! NANQ
>>
>> ...but the multiplie logical operator not working.
>> Can you help
>> GlanPlon
>
> You don't say what "not working" means, but my first guess is that the variables a and b are not
both scalars. You may need to re-cast your problem using a vector operation, a WHERE or a
loop.
>
> Jim P
```

Subject: Re: Logical operators
Posted by [Jim Pendleton](#) on Sun, 13 Nov 2016 16:39:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Sunday, November 13, 2016 at 9:18:27 AM UTC-7, Glan wrote:

```
> Thanks
> By not working, I mean the conditions are not applied and I get C=!NANQ everywhere.
> Can logical operators be combined in idl or not.
> Glan
```

```
>
```

> On Sunday, November 13, 2016 at 10:40:46 AM UTC-5, Jim P wrote:

```
>> On Sunday, November 13, 2016 at 7:30:50 AM UTC-7, Glan wrote:
```

```
>>> Hi
```

```
>>> I am trying to do
```

```
>>> if( ( a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50)) then begin
```

```
>>> c=! NANQ
```

```
>>>
```

```
>>> ...but the multiplie logical operator not working.
```

```
>>> Can you help
```

```
>>> GlanPlon
```

```
>>
```

>> You don't say what "not working" means, but my first guess is that the variables a and b are not both scalars. You may need to re-cast your problem using a vector operation, a WHERE or a loop.

```
>>
```

```
>> Jim P
```

My new guess is that A and B always satisfy the IF statement for your data. What combination of "a" and "b" is evaluated to TRUE that you expect to evaluate to FALSE?

Jim P

Subject: Re: Logical operators

Posted by [Phillip Bitzer](#) on Mon, 14 Nov 2016 03:05:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Sunday, November 13, 2016 at 8:30:50 AM UTC-6, Glan wrote:

```
> Hi
> I am trying to do
> if( ( a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50)) then begin
> c=! NANQ
>
> ...but the multiplie logical operator not working.
> Can you help
> GlanPlon
```

First, and just to clarify for others, "and" and "or" or not, strictly speaking, logical operators (&&, ||). They are bitwise. Here, they essentially work the same as the logical ops, since the result of the relational operators (lt,gt, etc.) lead to ones or zeros. But, the logical operators have the added benefit of "short-circuiting," so that `1 || _some_expression` is true, without evaluating `_some_expression`. Similarly, `0 && _some_expression` is false, without evaluating `_some_expression`.

With that said, a simple test shows the code does not always evaluate to true:

```
IDL> a = 0 & b=0
IDL> ( (a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50))
0
IDL> a = 0 & b=51
IDL> ( (a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50))
1
IDL> a = 51 & b=0
IDL> ( (a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50))
1
IDL> a = 51 & b=51
IDL> ( (a gt 50) and ( b lt 50) ) or ((a lt 50) and ( b gt 50))
0
```

(Your code seems to be missing a parentheses, BTW.)
