Subject: MOD operator
Posted by Theo Brauers on Tue, 17 Oct 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi experts.

I was wondering if the result of the MOD operator in IDL changed
from previous versions (4.x, 5.0) to the current version (5.3.1).
Now the output is:

```
IDL> PRINT, (FINDGEN(8)-4.) MOD 3
    -1.00000     0.000000     -2.00000     -1.00000     0.000000
     1.00000      2.00000     0.000000
```

When I programmed a function long ago I used the MOD operator
expecting the output

```
IDL> PRINT, (FINDGEN(8)-4.) MOD 3
     2.00000     0.000000      1.00000      2.00000     0.000000
     1.00000      2.00000     0.000000
```

I used MOD to get rid number lower than zero and higher than
23.999999 when calculating the hours of the day for example.

Any help or tests with old versions are appreciated. Thanks

Theo

--
---
Dr. Theo Brauers
Institut fuer Atmosphaerische Chemie (ICG-3)
Forschungszentrum Juelich
52425 JUELICH, Germany
Tel. +49-2461-61-6646  Fax. +49-2461-61-5346
http://www.kfa-juelich.de/icg/icg3/MITARBEITER/th.brauers.ht ml

Subject: Re: MOD operator
Posted by Theo Brauers on Wed, 18 Oct 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Thanks for your help. In the meantime I found (with help from Franz Rohrer
and Markus Fuehrer) that there is a difference between
 a MOD b = a-INT(a/b)*b
and
 modulo(a,b) = a-FLOOR(a/b)*b

I wrote an function calc_modulo
---
```
FUNCTION calc_modulo, a, b
 RETURN, a - FLOOR(a/b)*b
END
```
---
which works fine for me.
Cheers Theo
>
> Theo Brauers wrote:
>>
>> I was wondering if the result of the MOD operator in IDL changed
>> from previous versions (4.x, 5.0) to the current version (5.3.1).
>> Now the output is:
>>
>> IDL> PRINT, (FINDGEN(8)-4.) MOD 3
>>  -1.00000  0.000000  -2.00000  -1.00000  0.000000
>>  1.00000  2.00000  0.000000
>>
>> When I programmed a function long ago I used the MOD operator
>> expecting the output
>>
>> IDL> PRINT, (FINDGEN(8)-4.) MOD 3
>>  2.00000  0.000000  1.00000  2.00000  0.000000
>>  1.00000  2.00000  0.000000
>>
>
> To the best of my knowledge, this is always the way IDL (and most other
> languages providing a MOD operator) has worked. If you wish to take the
> mod of a negative number , the result will be a negative number between
> -(n+1) and 0. A positive input yields a positive result. So when we're
> trying to limit a value to between 0 and 2*pi, we often end up with code
> that looks something like:
>
> x = ((y mod (2*pi)) + 2*pi) mod (2*pi)
>
> Hope this helps.
>
> Phillip

--
---
Dr. Theo Brauers
Institut fuer Atmosphaerische Chemie (ICG-3)
Forschungszentrum Juelich
52425 JUELICH, Germany
Tel. +49-2461-61-6646  Fax. +49-2461-61-5346