Subject: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Wed, 18 Apr 2007 22:48:28 GMT
View Forum Message <> Reply to Message

Hi Guys,
There's something I can not explain to myself, so maybe someone can enlighten me?

```
print,fix(4.70*100)
     469
```

and also:

```
print,string(4.70*100,format='(i3)')
469
```

While, everything else that came into my head to try was OK like:

```
print,fix(5.70*100)
     570
 print,fix(3.70*100)
     370
 print,fix(4.60*100)
     460
print,string(4.60*100,format='(i3)')
460
...
etc..
```

Thanks in advance!
Cheers.
Boryana

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 19:43:13 GMT
View Forum Message <> Reply to Message

>   ROUND(469.999)
> (or vice versa)

Yes, this was the case.
I was pretty sure that converiting to integer means rounding to the nearest integer,
which is obviously not the case.

It took me a while to get it but well... bad for me.

Cheers
Boryana

Subject: Re: fix(4.70*100) is... 469
Posted by Paul Van Delst[1] on Thu, 19 Apr 2007 20:44:04 GMT

View Forum Message <> Reply to Message

b_efremova@yahoo.com wrote:
>>   ROUND(469.999)
>> (or vice versa)
>
> Yes, this was the case.
> I was pretty sure that converiting to integer means rounding to the
> nearest integer,
> which is obviously not the case.
>
> It took me a while to get it but well... bad for me.

Oh, I think "good for you!" is more appropriate. You're not likely to forget the lesson learned. That's a good thing, right? A knowledgeable fellow in another newsgroup I read has the following .sig:

Good judgement comes from experience;
experience comes from bad judgement.
   -- Mark Twain

We've all been there. (And the honest ones amongst us will also admit we still go there not infrequently :o)

cheers,

paulv

--
Paul van Delst   Ride lots.
CIMSS @ NOAA/NCEP/EMC   Eddy Merckx

Subject: Re: fix(4.70*100) is... 469
Posted by Paul Van Delst[1] on Fri, 20 Apr 2007 14:27:51 GMT

View Forum Message <> Reply to Message

Qing wrote:
> On Apr 20, 3:44 am, b_efrem...@yahoo.com wrote:

> Lets continue the hunt... it may not be just fun. Is is possible that
> the difference between 469 and 470 could end up sending a satellite
> off its track :-((

Sure. Multiply either number by 1AU and the difference is not insignificant

> Really, I am serious! :-)))

Me too. But the fault would lie with the spacecraft system programmer, not the language (be it IDL or something else).

Actually, the fault would lie with the programmers' supervisor/manager for a) assigning an inexperienced person to produce the code and/or b) for not establishing useful QC/QA processes that should catch these sorts of naive coding errors.

cheers,

paulv

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: fix(4.70*100) is... 469
Posted by James Kuyper on Fri, 20 Apr 2007 16:26:07 GMT

View Forum Message <> Reply to Message

Qing wrote:
...
> (1) It is not a problem about just the FIX function:
>
> IDL> print, floor(4.7*100)
>   469
>
> (2) Can we always use ROUND instead of FIX or FLOOR? Then why the hell
> to have FIX and FLOOR to get confused?

No, you cannot always use ROUND instead of FIX or FLOOR, because, believe it or not, in some contexts the different result returned by FLOOR is more appropriate for what needs to be done. When I use FLOOR(x), I want any value that is greater than or equal to 469, and less than 470, to result in a return value of 469. The expression 4.7*100 has a value (at least on this hardware) which is less than 470. The number 470 can generally be represented exactly, but 4.7 cannot, and 4.7*100 is therefore extremely unlikely to calculate a value of exactly 470.

I use FLOOR far more often than ROUND, because most of the contexts where I might otherwise want to use ROUND are already handled correctly inside the formatted IO routines.

> (3) A precision issue? definitely need to read again at
> http://www.dfanning.com/math_tips/sky_is_falling.html .

> "There's nothing worse that trying to debug code and discovering weird
> results are related to the precision of the represetation" - it can
> also be fun!
> But what about:
>
> IDL> print, fix(100D*4.7)   ============= it can still be argued
> as a precision issue as long as you use 4.7 as a example !!!
>   469

It is an inherent, consequence of the finite precision of the binary
floating point representation that it cannot represent 4.7 exactly;
the best it can do is a number either slightly larger or slightly
smaller than 4.7. If your code depends upon that difference, you
shouldn't be using floating point representations; use fixed pointer
representations instead.

> (4) "... maybe apart from an insidious compiler bug, but that would
> never happen with IDL!"
> does the problem happen in just IDL (on Windows, Lenux, MaxOS, ...)?
> Do we have the same problem in C/C++, FORTRAN or even BASIC?

Yes, this happens in every computer language that allows decimal
fractions to be stored in binary format.

> Lets continue the hunt... it may not be just fun. Is is possible that
> the difference between 469 and 470 could end up sending a satellite
> off its track :-((

Only when code is written which handles incorrectly the
characteristics of floating point data. Which, admittedly, is not rare.

Subject: Re: fix(4.70*100) is... 469
Posted by James Kuyper on Fri, 20 Apr 2007 16:31:20 GMT

View Forum Message <> Reply to Message

Paolo Grigis wrote:
> David Fanning wrote:
>> Jean H. writes:
>>
>>> What is being displayed is a bit different than what is being stored..
>>> IDL> print, 470.0 - (4.70*100)
>>>   3.05176e-005
>>
>> I guess I would argue that what is being displayed is EXACTLY
>> what is being stored:
>
> Is it?

> I always wondered if the garbage after the 16th digit does
> correspond indeed to the exact decimal representation of the
> double stored in memory in binary format, or is just a meaningless
> side product of the binary to decimal conversion...
>
> IDL> print,!DPi,format='(f68.64)'
> 3.1415926535897931159979634685441851615905761718750000000000 000000

The fractional part of a floating point number represented in binary
format can always be written as

>
>
> Ciao,
> Paolo
>
>
>>
>> IDL> print, 470, format='(f18.14)'
>> 470.00000000000000
>> IDL> print, 4.70*100, format='(f18.14)'
>> 469.99996948242187
>> IDL> print, 470.00000000000000D - 469.99996948242187D, format='(f18.14)'
>> 0.00003051757813
>>
>> Cheers,
>>
>> David

Subject: Re: fix(4.70*100) is... 469
Posted by James Kuyper on Fri, 20 Apr 2007 16:43:36 GMT
View Forum Message <> Reply to Message

Paolo Grigis wrote:
> David Fanning wrote:
...
>> I guess I would argue that what is being displayed is EXACTLY
>> what is being stored:
>
> Is it?
> I always wondered if the garbage after the 16th digit does
> correspond indeed to the exact decimal representation of the
> double stored in memory in binary format, or is just a meaningless
> side product of the binary to decimal conversion...
>
> IDL> print,!DPi,format='(f68.64)'

> 3.1415926535897931159979634685441851615905761718750000000000 000000

The fractional part of a floating point number represented in binary
format can always be written as an integer numerator divided by a
larger power of 2:

numerator

2^n

It can therefore be represented exactly by a decimal fraction as

numerator*5^n

10^n

The are known algorithms for converting a floating point number to a
decimal string which will produce precisely that value, if you tell
them to use enough digits. I don't know any simple way of testing
this, but I would assume that IDL uses such an algorithm.

The extra digits won't make the decimal string a better representation
of pi, but they will make it an accurate representation of the
computer's best binary approximation to pi.

Subject: Re: fix(4.70*100) is... 469
Posted by Bruce Bowler on Fri, 20 Apr 2007 16:47:31 GMT

View Forum Message <> Reply to Message

> It is an inherent, consequence of the finite precision of the binary
> floating point representation

If only we didn't have pinkies and baby toes.

Bruce

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Fri, 20 Apr 2007 17:30:56 GMT

View Forum Message <> Reply to Message

In article <pan.2007.04.20.16.47.31@bigelow.org>, Bruce Bowler <bbowler@bigelow.org> writes:
>> It is an inherent, consequence of the finite precision of the binary
>> floating point representation
>
> If only we didn't have pinkies and baby toes.

>
Ahh, legacy issues:-)

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"