Subject: Re: fix(4.70*100) is... 469
Posted by Jean H. on Wed, 18 Apr 2007 23:48:08 GMT
View Forum Message <> Reply to Message

> Hooboy! Haven't had one of these in awhile. :-)
>
> http://www.dfanning.com/math_tips/sky_is_falling.html
>
> Cheers,
>
> David

It seems a bit weirder though...

```
IDL> print, 470.0 - (4.70*100)
  3.05176e-005
```

which is bigger than the smallest float

```
IDL> print, (machar()).eps
  1.19209e-007
```

.... so how can it be the float accuracy problem if the difference between the expected and the real value is 256 times bigger than the float error?

Jean

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 00:12:04 GMT
View Forum Message <> Reply to Message

b_efremova@yahoo.com writes:

```
> 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..
```

Hooboy! Haven't had one of these in awhile. :-)

   http://www.dfanning.com/math_tips/sky_is_falling.html

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 00:14:31 GMT

View Forum Message <> Reply to Message

In article <f06ane$21t$1@news.ucalgary.ca>, "Jean H." <jghasban@DELTHIS.ucalgary.ANDTHIS.ca> writes:

```
>>  Hooboy! Haven't had one of these in awhile. :-)
>>
>>     http://www.dfanning.com/math_tips/sky_is_falling.html
>>
>>  Cheers,
>>
>>  David
>
> It seems a bit weirder though...
>
> IDL> print, 470.0 - (4.70*100)
>   3.05176e-005
>
> which is bigger than the smallest float
>
> IDL> print, (machar()).eps
>   1.19209e-007
>
> .... so how can it be the float accuracy problem if the difference
> between the expected and the real value is 256 times bigger than the
```

> float error?
>

Careful here. The smallest float provides relative accuracy, meaning
the difference between exact and stored value X doesn't exceed
X*(machar()).eps. This is well satisfied here.

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

Subject: Re: fix(4.70*100) is... 469
Posted by Jean H. on Thu, 19 Apr 2007 00:58:08 GMT

View Forum Message <> Reply to Message

>> .... so how can it be the float accuracy problem if the difference
>> between the expected and the real value is 256 times bigger than the
>> float error?
>>
> Careful here. The smallest float provides relative accuracy, meaning
> the difference between exact and stored value X doesn't exceed
> X*(machar()).eps. This is well satisfied here.
>
> Mati Meron | "When you argue with a fool,
> meron@cars.uchicago.edu | chances are he is doing just the same"

I don't understand why one should multiply epsilon by X .... why would
the acceptable difference between expect and real value be a function of
the value? ... X = 100.0 Y = 900.0 .... they both have the same number
of significant digits, so why would the max acceptable difference be

```
IDL> print, 100.0 * epsilon
  1.19209e-005
IDL> print, 900.0 * epsilon
   0.000107288
```

Also, if one must really multiply epsilon by X, does it mean that there
is an error on http://www.dfanning.com/math_tips/razoredge.html, at the
last line of the page?

```
IDL> print,abs(0.9 - (0.6+0.3)) lt (machar()).eps
```

should it be

```
IDL> print,abs(0.9 - (0.6+0.3)) lt 0.9 * (machar()).eps
```

???

I admit to be lost on this issue... and it scares me as I might have to
check/change all my codes!!!!!

Do you have a reference at hand on this?

Thanks,
Jean

Subject: Re: fix(4.70*100) is... 469

Posted by Qing on Thu, 19 Apr 2007 04:31:47 GMT

View Forum Message <> Reply to Message

On Apr 19, 8:48 am, b_efrem...@yahoo.com wrote:

```
> 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
```

Look at that!

```
IDL> print, double(4.7)
       4.6999998
```

Subject: Re: fix(4.70*100) is... 469

Posted by Qing on Thu, 19 Apr 2007 04:36:32 GMT

View Forum Message <> Reply to Message

On Apr 19, 2:31 pm, Qing <c...@bigpond.net.au> wrote:

```
>  On Apr 19, 8:48 am, b_efrem...@yahoo.com wrote:
>
>
>
>
>
>>  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
>
>  Look at that!
>  IDL> print, double(4.7)
>        4.6999998- Hide quoted text -
>
>  - Show quoted text -
```

And more ...

```
IDL> print, double(9.4)
      9.3999996
IDL> print, double(2.35)
      2.3499999
IDL> print, double(18.8)
      18.799999
IDL> print, double(18.9)
      18.900000
IDL> print, double(18.7)
      18.700001
```

...

Subject: Re: fix(4.70*100) is... 469
Posted by Qing on Thu, 19 Apr 2007 04:38:15 GMT
View Forum Message <> Reply to Message

```
On Apr 19, 2:36 pm, Qing <c...@bigpond.net.au> wrote:
>  On Apr 19, 2:31 pm, Qing <c...@bigpond.net.au> wrote:
>
>
>
>
>
>>  On Apr 19, 8:48 am, b_efrem...@yahoo.com wrote:
>
>>>  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
>
>>  Look at that!
>>  IDL> print, double(4.7)
>>         4.6999998- Hide quoted text -
>
>>  - Show quoted text -
>
>  And more ...
>  IDL> print, double(9.4)
>        9.3999996
>  IDL> print, double(2.35)
>        2.3499999
>  IDL> print, double(18.8)
>        18.799999
>  IDL> print, double(18.9)
>        18.900000
>  IDL> print, double(18.7)
>        18.700001
>  ...- Hide quoted text -
>
>  - Show quoted text -
```

```
IDL> print, double(4.7*1)
       4.6999998
IDL> print, double(4.7*10)
       47.000000
IDL> print, double(4.7*100)
       469.99997
IDL> print, double(4.7*1000)
       4700.0000
```

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 05:56:23 GMT

View Forum Message <> Reply to Message

In article <f06eqm$412$1@news.ucalgary.ca>, "Jean H."
<jghasban@DELTHIS.ucalgary.ANDTHIS.ca> writes:

```
>>>  .... so how can it be the float accuracy problem if the difference
>>>  between the expected and the real value is 256 times bigger than the
>>>  float error?
>>>
>>  Careful here.  The smallest float provides relative accuracy, meaning
>>  the difference between exact and stored value X doesn't exceed
>>  X*(machar()).eps.  This is well satisfied here.
>>
>>  Mati Meron                    | "When you argue with a fool,
>>  meron@cars.uchicago.edu       |  chances are he is doing just the same"
>
> I don't understand why one should multiply epsilon by X .... why would
> the acceptable difference between expect and real value be a function of
> the value? ... X = 100.0  Y = 900.0  .... they both have the same number
> of significant digits, so why would the max acceptable difference be
> IDL> print, 100.0 * epsilon
>   1.19209e-005
> IDL> print, 900.0 * epsilon
```

>   0.000107288
>
>
> Also, if one must really multiply epsilon by X, does it mean that there
> is an error on http://www.dfanning.com/math_tips/razoredge.html, at the
> last line of the page?
>
> IDL> print,abs(0.9 - (0.6+0.3)) lt (machar()).eps
>
> should it be
> IDL> print,abs(0.9 - (0.6+0.3)) lt 0.9 * (machar()).eps
>
> ???
>
> I admit to be lost on this issue... and it scares me as I might have to
> check/change all my codes!!!!!
>
> Do you have a reference at hand on this?
>

Consider what "same number of significant digits mean. For example, consider that 1.23456*10^20 and 1.23456*10^(-20) have same number of significant digits.

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

Subject: Re: fix(4.70*100) is... 469
Posted by Paolo Grigis on Thu, 19 Apr 2007 08:13:53 GMT

View Forum Message <> Reply to Message

b_efremova@yahoo.com wrote:
> 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
>
```

Personally, I think "fix" should be used with some care,
because the documentation is a bit vague on how the
conversion to integer type is done, and the result does
depend on the input type. For numerical values, I think
that usage of "round", "ceil" or "floor" is safer (but be
aware that they produce long integers if the input
is large enough, but this is mostly what you want anyway).

Ciao,
Paolo

Subject: Re: fix(4.70*100) is... 469
Posted by KRDean on Thu, 19 Apr 2007 13:41:25 GMT

View Forum Message <> Reply to Message

Oh, we can have fun with this.

Starting with Qing reply....

```
 IDL> print, double(4.7)
       4.6999998
```

How about this...

```
IDL> print, 4.7D
       4.7000000
```

Or, even this...

```
IDL> print, DOUBLE(4.7E1)
       47.000000
```

Even stranger ....

```
IDL> print, DOUBLE(STRING(4.7,FORMAT='(F0)'))
       4.7000000
```

Kelly Dean
Fort Collins, CO

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 14:32:59 GMT
View Forum Message <> Reply to Message

Sorry Guys, I should have made myself clearer.
I'm afraid David this is not actually the question you describe in
your article.
and I do not expect better accuracy than I provide.

There is nothing wrong here with the floating point accuracy.
print,4.700*100.00
470.000

It is the conversion to integer (I imagine) which makes no sence.

print,fix(4.700*100.00)
469
also (which is what I really needed)

print,string(4.700*100.00,format='(i3)')
469

Cheers
Boryana

> Hooboy! Haven't had one of these in awhile. :-)
>
> http://www.dfanning.com/math_tips/sky_is_falling.html
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 14:41:21 GMT

View Forum Message <> Reply to Message

Sorry Guys, I should have made myself clearer.
I'm afraid David this is not actually the question you describe in your article.
and I do not expect better accuracy than I provide.

There is nothing wrong here with the floating point accuracy.

```
print,4.700*100.00
470.000
```

It is the conversion to integer (I imagine) which makes no sence.

```
print,fix(4.700*100.00)
469
```

also (which is what I really needed)

```
print,string(4.700*100.00,format='(i3)')
469
```

Cheers
Boryana

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 14:42:33 GMT

View Forum Message <> Reply to Message

mmeron@cars3.uchicago.edu writes:

> Consider what "same number of significant digits mean. For example,
> consider that 1.23456*10^20 and 1.23456*10^(-20) have same number of
> significant digits.

Alright, you have completely lost me here. Can you
expand this argument just a wee bit more? :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469

Posted by David Fanning on Thu, 19 Apr 2007 14:48:42 GMT

View Forum Message <> Reply to Message

kBob writes:

```
>   Oh, we can have fun with this.
>
>   Starting with Qing reply....
>
>   IDL> print, double(4.7)
>         4.6999998
>
>    How about this...
>
>  IDL> print, 4.7D
>         4.7000000
>
>    Or, even this...
>
>  IDL> print, DOUBLE(4.7E1)
>         47.000000
>
>   Even stranger ....
>
>  IDL> print, DOUBLE(STRING(4.7,FORMAT='(F0)'))
>         4.7000000
```

Well, these are all explained in this article:

   http://www.dfanning.com/math_tips/double.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469

Posted by Paolo Grigis on Thu, 19 Apr 2007 15:02:43 GMT

View Forum Message <> Reply to Message

At the risk of reiterating what I already stated earlier:

fix truncates your number when converting it to integer.

Compare:

```
IDL> print,4.999999
      5.00000
IDL> print,fix(4.999999)
       4
```

You don't like this behavior? Fine, just use round instead:

```
IDL> print,round(4.999999)
           5
```

Ciao,
Paolo

b_efremova@yahoo.com wrote:
> Sorry Guys, I should have made myself clearer.
> I'm afraid David this is not actually the question you describe in
> your article.
> and I do not expect better accuracy than I provide.
>
> There is nothing wrong here with the floating point accuracy.
> print,4.700*100.00
> 470.000
>
> It is the conversion to integer (I imagine) which makes no sence.
>
> print,fix(4.700*100.00)
> 469
> also (which is what I really needed)
>
> print,string(4.700*100.00,format='(i3)')
> 469
>
> Cheers
> Boryana
>
>> Hooboy! Haven't had one of these in awhile. :-)
>>
>> http://www.dfanning.com/math_tips/sky_is_falling.html
>>
>> Cheers,
>>
>> David

>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>

Subject: Re: fix(4.70*100) is... 469
Posted by eddedmondson on Thu, 19 Apr 2007 15:13:23 GMT
View Forum Message <> Reply to Message

b_efremova@yahoo.com wrote:
> Sorry Guys, I should have made myself clearer.
> I'm afraid David this is not actually the question you describe in
> your article.
> and I do not expect better accuracy than I provide.
>
> There is nothing wrong here with the floating point accuracy.
> print,4.700*100.00
> 470.000
>

People have posted similar stuff where doubles have been brought in. It's perhaps clearer to post an example where doubles stay out of it:

```
IDL> print, format='(F9)', 4.7
4.6999998
```

You're not starting with all of your 4.7.

--
Edd

Subject: Re: fix(4.70*100) is... 469
Posted by Christopher Thom on Thu, 19 Apr 2007 15:21:56 GMT
View Forum Message <> Reply to Message

Quoth b_efremova@yahoo.com:

> Sorry Guys, I should have made myself clearer.
> I'm afraid David this is not actually the question you describe in
> your article.
> and I do not expect better accuracy than I provide.
>

> There is nothing wrong here with the floating point accuracy.
> print,4.700*100.00
> 470.000
>
> It is the conversion to integer (I imagine) which makes no sence.
>
> print,fix(4.700*100.00)
> 469
> also (which is what I really needed)

No. Read the article again...and the one on double precision...it is exactly what is described there. You have provided IDL with a number that has 8 decimal places of precision. 4.7 is really somewhere between 4.6999999 - 4.7000001, but cannot be precisely represented. i.e.

```
IDL> print,4.7
     4.70000
IDL> print,4.7,f='(f18.16)'
4.6999998092651367
```

The important point is that converting the *actual number as represented in the computer* to an integer, is NOT converting the number you *think* is represented in the computer.

So...if you take the number that is actually in IDL...move the decimal place 2 places to the right, you get

```
IDL> print,4.7*100,f='(f18.14)'
469.99996948242188
```

Now chop off every thing after the decimal place (which is what fix() does)...and 469 is a prefectly reasonable answer to the question you asked. If you want a better answer, you need to ask a better question :-)

I can't speak as to exactly how the conversion to integers happens within the string() command you gave, but I imagine it's probably the same.

cheers
chris

Subject: Re: fix(4.70*100) is... 469
Posted by Jean H. on Thu, 19 Apr 2007 15:35:14 GMT

View Forum Message <> Reply to Message

> There is nothing wrong here with the floating point accuracy.
> print,4.700*100.00
> 470.000

What is being displayed is a bit different than what is being stored..

```
IDL> print, 470.0 - (4.70*100)
  3.05176e-005
```

Jean

>
> It is the conversion to integer (I imagine) which makes no sence.
>
> print,fix(4.700*100.00)
> 469
> also (which is what I really needed)
>
> print,string(4.700*100.00,format='(i3)')
> 469
>
> Cheers
> Boryana
>

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 16:35:48 GMT

View Forum Message <> Reply to Message

Christopher Thom writes:

> No. Read the article again...and the one on double precision...it is
> exactly what is described there

Thank you. Excellent explanation! :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 16:50:36 GMT

View Forum Message <> Reply to Message

In article <MPG.20912dc2d706b423989f47@news.frii.com>, David Fanning
<news@dfanning.com> writes:
> mmeron@cars3.uchicago.edu writes:
>
>> Consider what "same number of significant digits mean. For example,
>> consider that 1.23456*10^20 and 1.23456*10^(-20) have same number of
>> significant digits.
>
> Alright, you have completely lost me here. Can you
> expand this argument just a wee bit more? :-)
>

Certainly. The floating number is stored as two parts, mantissa and power (for the garden variety float you've 24 bits for the mantissa and 8 for the power). The mantissa specifies the significant digits, which are then multiplied by the appropriate power. The storage is binary, of course, but for the purpose of this argument we may look at decimal. So, if you store, say, 7 significant digits, your number is of the form 0.abcdefg * 10^p, where a...f are digits between 0 and 9. If you take two numbers such that their true (as opposed to stored) expansion has same first 7 significant digits while differing at the 8th, they'll be stored as same number. So, roughly, one can say that the accuracy of the stored number is 0.00000005 *10^p (note, 7 zeroes for the significant digits, then half the maximum for the next). So, the storage error, for fixed number of decimal places, is relative, not absolute, it is around 0.00000005/0.abcdefg. As the magnitude of the number grows, so does the error. As you can see in the following sequence

```
IDL> print, 1 + 1e-8 - 1
    0.000000
IDL> print, 1e4 + 1e-4 - 1e4
    0.000000
IDL> print, 1e8 + 1 - 1e8
    0.000000
IDL> print, 1e28 + 1e20 - 1e20
 1.00000e+028
IDL> print, 1e28 + 1e20 - 1e28
    0.000000
```

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

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 16:54:46 GMT
View Forum Message <> Reply to Message

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:

```
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
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 17:00:07 GMT

View Forum Message <> Reply to Message

In article <1176993178.932710.145030@p77g2000hsh.googlegroups.com>,
b_efremova@yahoo.com writes:

```
> Sorry Guys, I should have made myself clearer.
> I'm afraid David this is not actually the question you describe in
> your article.
> and I do not expect better accuracy than I provide.
>
> There is nothing wrong here with the floating point accuracy.
> print,4.700*100.00
> 470.000
>
> It is the conversion to integer (I imagine) which makes no sence.
>
> print,fix(4.700*100.00)
> 469
> also (which is what I really needed)
>
> print,string(4.700*100.00,format='(i3)')
```

> 469
>
Nah, it is a floating point accuracy issue. Remember, the output of
print is *not* necessarily the number stored in memory, it is just
said number rounded to some default number of decimal places (cost me
many sleepless nights, this one). So, indeed

```
IDL> print, 4.7*100
      470.000
```

But, lets take a look at more decimal places

```
IDL> print, 4.7*100, format = '(f12.8)'
469.99996948
```

Since FIX always picks the integer part throwing away the fraction
(i.e. always rounds down) this becomes 469. You'll avoid this problem
if you'll use ROUND instead of FIX.

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

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 17:03:55 GMT
View Forum Message <> Reply to Message

Thanks Chris,
I did,
I'll try a better quesstion then.
I totally agree that 470 is something between 469.9999 something and
470.00000 something
Here comes my question then:

```
print,469.9999,format='(f9.3)'
470.000

....
print,469.9999,format='(f9.0)'
470.

print,469.9999,format='(i3)'
469.
```

So converting to integer DOES NOT keep the number the same with the
give accuracy.
Instead it just truncates.

Sorry it took me so long to explain myself.

And thanks Paolo, I use round, and all works fine.

Cheers
Boryana

Christopher Thom wrote:
> Quoth b_efremova@yahoo.com:
>
>> Sorry Guys, I should have made myself clearer.
>> I'm afraid David this is not actually the question you describe in
>> your article.
>> and I do not expect better accuracy than I provide.
>>
>> There is nothing wrong here with the floating point accuracy.
>> print,4.700*100.00
>> 470.000
>>
>> It is the conversion to integer (I imagine) which makes no sence.
>>
>> print,fix(4.700*100.00)
>> 469
>> also (which is what I really needed)
>
> No. Read the article again...and the one on double precision...it is
> exactly what is described there. You have provided IDL with a number that
> has 8 decimal places of precision. 4.7 is really somewhere between
> 4.6999999 - 4.7000001, but cannot be precisely represented. i.e.
>
> IDL> print,4.7
>   4.70000
> IDL> print,4.7,f='(f18.16)'
> 4.6999998092651367
>
> The important point is that converting the *actual number as represented
> in the computer* to an integer, is NOT converting the number you *think*
> is represented in the computer.
>
> So...if you take the number that is actually in IDL...move the decimal
> place 2 places to the right, you get
>
> IDL> print,4.7*100,f='(f18.14)'
> 469.99996948242188
>

> Now chop off every thing after the decimal place (which is what fix()
> does)...and 469 is a prefectly reasonable answer to the question you
> asked. If you want a better answer, you need to ask a better question :-)
>
> I can't speak as to exactly how the conversion to integers happens within
> the string() command you gave, but I imagine it's probably the same.
>
> cheers
> chris

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 17:44:32 GMT
View Forum Message <> Reply to Message

I'll try one more time , see if it works.

Here is my initial statement:
When I handle a floating point number, 470.000
it better stay the same number with the precision I use it.

It may well be 469.9999999999999999999999 and this is
perfectly fine with me if when used with a floating point precision
it is rounded to 470.000

OK, so if I do something, like
converting 470.00 to double, I have no right to complain when
the number I get is 469.99999999999999435462346 with any
arbitrary numbers in the digits exceeding the precision I had
provided.

On the other side, I would expect when I use this number with the
provided precision or lower to matter if I had given the number
470.000 or 469.999.

In short, when converting my number to something of lower precision
like integer
I would expect the number to be rounded.

but the integer of 469.9999 is 469. which is not true.

I also had the wrong assumption that FIX will FIRST round the number
to the precision I'm
working with and THEN truncate.
Now when I think of this, it is not very intellicgent assumption.
Well, I did it.
On the other hand I stilll think that

print,469.9999,format='(i3)'
should be
470

but it is 469 insted.
What is wrong about this assumption of mine?
Cheers
Boryana

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 18:13:54 GMT

View Forum Message <> Reply to Message

Thanks, Incredibly helpful.

Do you mind to add a line explaining why

print,479.9999,format='(i3)'
479

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 18:15:40 GMT

View Forum Message <> Reply to Message

In article <MPG.20916bc4bfdc7a43989f4c@news.frii.com>, David Fanning
<news@dfanning.com> writes:
> David Fanning writes:
>
>> I don't think there is anything "wrong" about your
>> assumption, except that it doesn't reflect reality.
>> This is how most of us go wrong with assumptions.
>> We assume something is so that ain't so. :-)
>
> Just to enlarge on this a bit, after poor typing
> skills, incorrect assumptions probably makes up
> the largest percentage of "bugs" in software. If
> you hear someone shouting "I don't f---ing
> believe this!", you can be almost certain an
> assumption is going to come crashing down.
>
> In general, there are better outcomes when our
> assumptions closely model reality than when they
> do not. In this particular case, if your assumption
> is "floating point math on computers can drive you
> crazy", then you are likely to make fewer mistakes

> than if your assumption is "computers always do
> what I expect them to do."
>
In other words, "an ounce of paranoia saves a pound of debugging".

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

Subject: Re: fix(4.70*100) is... 469
Posted by b_efremova@yahoo.com on Thu, 19 Apr 2007 18:47:57 GMT
View Forum Message <> Reply to Message

> In other words, "an ounce of paranoia saves a pound of debugging".

Thanks guys, all this was very helpful to me!
Sorry I bored you.
Cheers
Boryana

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 18:52:16 GMT
View Forum Message <> Reply to Message

b_efremova@yahoo.com writes:

> What is wrong about this assumption of mine?

I don't think there is anything "wrong" about your
assumption, except that it doesn't reflect reality.
This is how most of us go wrong with assumptions.
We assume something is so that ain't so. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by Paul Van Delst[1] on Thu, 19 Apr 2007 19:06:48 GMT

b_efremova@yahoo.com wrote:
> I'll try one more time , see if it works.
>
> Here is my initial statement:
> When I handle a floating point number, 470.000
> it better stay the same number with the precision I use it.
>
> It may well be 469.9999999999999999999999 and this is
> perfectly fine with me if when used with a floating point precision
> it is rounded to 470.000
>
> OK, so if I do something, like
> converting 470.00 to double, I have no right to complain when
> the number I get is 469.99999999999999435462346 with any
> arbitrary numbers in the digits exceeding the precision I had
> provided.
>
> On the other side, I would expect when I use this number with the
> provided precision or lower to matter if I had given the number
> 470.000 or 469.999.

What do *you* mean by "provided precision"?

> In short, when converting my number to something of lower precision
> like integer
> I would expect the number to be rounded.

What you expect and what actually happens is demonstrably (and reproducably) different. There are several ways to "convert" a floating point number to an integer. Rounding is just one way.

> but the integer of 469.9999 is 469. which is not true.
>
> I also had the wrong assumption that FIX will FIRST round the number
> to the precision I'm
> working with and THEN truncate.
> Now when I think of this, it is not very intellicgent assumption.
> Well, I did it.

Ah, well. We are now in the realm of the lessons learnt in the school of hard knocks. :o) It can be a royal pain in the rear end, but it can be good way of learning things -- we tend not to forget the associated blood, sweat and tears.

> On the other hand I stilll think that
>
> print,469.9999,format='(i3)'
> should be

> 470
>
> but it is 469 insted.
> What is wrong about this assumption of mine?

It's simply another syntax for

```
IDL> print, INT(469.999)
       469
```

Computers have zero intelligence - you have to ask them to do *exactly* what you want.
There's no way for the circuitry to divine that while you actually said something like

```
  INT(469.999)
```

you really meant

```
  ROUND(469.999)
```

(or vice versa)

Many a program has crashed in a big hairy heap on the floor because of this type of assumption.

Have a read of:
  http://docs.sun.com/source/806-3568/ncg_goldberg.html

and check out
  http://tinyurl.com/2wnggj

from your library.

cheers,

paulv

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

Subject: Re: fix(4.70*100) is... 469
Posted by David Fanning on Thu, 19 Apr 2007 19:07:04 GMT

View Forum Message <> Reply to Message

David Fanning writes:

> I don't think there is anything "wrong" about your
> assumption, except that it doesn't reflect reality.
> This is how most of us go wrong with assumptions.
> We assume something is so that ain't so. :-)

Just to enlarge on this a bit, after poor typing

skills, incorrect assumptions probably makes up
the largest percentage of "bugs" in software. If
you hear someone shouting "I don't f---ing
believe this!", you can be almost certain an
assumption is going to come crashing down.

In general, there are better outcomes when our
assumptions closely model reality than when they
do not. In this particular case, if your assumption
is "floating point math on computers can drive you
crazy", then you are likely to make fewer mistakes
than if your assumption is "computers always do
what I expect them to do."

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: fix(4.70*100) is... 469
Posted by Qing on Fri, 20 Apr 2007 06:43:43 GMT

View Forum Message <> Reply to Message

On Apr 20, 3:44 am, b_efrem...@yahoo.com wrote:
> I'll try one more time , see if it works.
>
> Here is my initial statement:
> When I handle a floating point number, 470.000
> it better stay the same number with the precision I use it.
>
> It may well be 469.9999999999999999999999 and this is
> perfectly fine with me if when used with a floating point precision
> it is rounded to 470.000
>
> OK, so if I do something, like
> converting 470.00 to double, I have no right to complain when
> the number I get is 469.99999999999999435462346 with any
> arbitrary numbers in the digits exceeding the precision I had
> provided.
>
> On the other side, I would expect when I use this number with the
> provided precision or lower to matter if I had given the number
> 470.000 or 469.999.

>
> In short, when converting my number to something of lower precision
> like integer
> I would expect the number to be rounded.
>
> but the integer of 469.9999 is 469. which is not true.
>
> I also had the wrong assumption that FIX will FIRST round the number
> to the precision I'm
> working with and THEN truncate.
> Now when I think of this, it is not very intellicgent assumption.
> Well, I did it.
> On the other hand I stilll think that
>
> print,469.9999,format='(i3)'
> should be
> 470
>
> but it is 469 insted.
> What is wrong about this assumption of mine?
> Cheers
> Boryana

Definately "sky is falling" introduces great fun and interests to us scientists! To continue ...

(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?

(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
```

(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?

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 :-((

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

Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Fri, 20 Apr 2007 07:33:51 GMT
View Forum Message <> Reply to Message

In article <1177051423.028405.260680@y80g2000hsf.googlegroups.com>, Qing
<csis@bigpond.net.au> writes:
> On Apr 20, 3:44 am, b_efrem...@yahoo.com wrote:
>> I'll try one more time , see if it works.
>>
>> Here is my initial statement:
>> When I handle a floating point number, 470.000
>> it better stay the same number with the precision I use it.
>>
>> It may well be 469.99999999999999999999999 and this is
>> perfectly fine with me if when used with a floating point precision
>> it is rounded to 470.000
>>
>> OK, so if I do something, like
>> converting 470.00 to double, I have no right to complain when
>> the number I get is 469.99999999999999435462346 with any
>> arbitrary numbers in the digits exceeding the precision I had
>> provided.
>>
>> On the other side, I would expect when I use this number with the
>> provided precision or lower to matter if I had given the number
>> 470.000 or 469.999.
>>
>> In short, when converting my number to something of lower precision
>> like integer
>> I would expect the number to be rounded.
>>
>> but the integer of 469.9999 is 469. which is not true.
>>
>> I also had the wrong assumption that FIX will FIRST round the number
>> to the precision I'm
>> working with and THEN truncate.
>> Now when I think of this, it is not very intellicgent assumption.
>> Well, I did it.

>> On the other hand I stilll think that
>>
>> print,469.9999,format='(i3)'
>> should be
>> 470
>>
>> but it is 469 insted.
>> What is wrong about this assumption of mine?
>> Cheers
>> Boryana
>
> Definately "sky is falling" introduces great fun and interests to us
> scientists! To continue ...
>
> (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?
>
Because they provide answers to different questions. Given a floating
point number X, you may be (depending on the problem you're dealing
with) interested in:

1) The closest integer approximation to X
2) The upper bound of all integers <= X
3) The lower bound of all integers >= X

ROUND provides the answer to the first question, FLOOR to the second,
CEIL to the third. So they're all useful. FIX is the worst choice in
general since it behaves like FLOOR for positives integers but like
CEIL for negative ones.

> (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
>
> (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?
>
> 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 :-((
>
> Really, I am serious! :-)))
> Qing
>
>
>
>

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

Subject: Re: fix(4.70*100) is... 469
Posted by Paolo Grigis on Fri, 20 Apr 2007 08:04:14 GMT
View Forum Message <> Reply to Message

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)'
    3.141592653589793115997963468544185161590576171875000000000 000000
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

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
```