
Subject: Re: print, long(1.0e10)

Posted by [Liam Gumley](#) on Wed, 04 Aug 1999 07:00:00 GMT

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

Gary Fu wrote:

> I got different results from the following example on SGI IRIX and PC
> Linux:
> a = 1.0e10
> b = long(a)
> print, b ; 2147483647 for IRIX, -2147483648 for Linux
>
> It looks like that IRIX and Linux handle it differently when assigning a
> floating value greater (or less) than the boundary of a long integer to
> a long variable. Should I always check the range of 'a' before
> assigning it to a 'long' variable to make the result consistent on IRIX
> and Linux ?

Gary,

I guess it depends on what you are trying to do. If you want to interpret the binary representation of 1.0e10 as a 32-bit signed LONG, then the syntax is

```
result = long(1.0e10, 0)
```

However if you want to represent 1.0e10 as an integer value, I think the only option (other than a range check) is to use IDL 5.2, which allows 64-bit signed and unsigned integers, e.g.

```
result = long64(1.0e10)
```

Cheers,
Liam.

--

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>
