
Subject: Re: inverse function of binary.pro
Posted by [David Fanning](#) on Wed, 30 Jun 2004 14:20:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

sebinjapan writes:

> I am using the function (best_)binary.pro available on David Fanning's web
> page:
> http://www.dfanning.com/misc_tips/binary_hex.html
>
> I would like to know if someone already wrote the inverse function of
> binary (a function that gives you the number corresponding to its binary
> representation)?
> Something like:
> IDL> print, inv_binary(binary(76.12))
> 76.12

For something quick and dirty, how about this:

```
FUNCTION Inverse_Binary, binaryNumber

  s = Size(binaryNumber, /Dimensions)
  bn = Reform(Long(binaryNumber), 8*s[1])
  len = N_Elements(bn)

  RETURN, Total(bn*2^Reverse(Indgen(len)))

END
```

Works for the two or three values I've tested. :-)

> btw: I think there is a minor error at the end of binary.pro. On little
> endian computer, it doesn't work with scalar of byte type. I think the
> last "if" test should be:
> if little_endian AND type NE 1 then \$
> instead of
> if little_endian then \$

Kevin Ivory gave me this little program, but I think you are right about it. I've updated my web page with the change.
Thanks.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: inverse function of binary.pro
Posted by [David Fanning](#) on Wed, 30 Jun 2004 14:27:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

```
> For something quick and dirty, how about this:  
>  
> FUNCTION Inverse_Binary, binaryNumber  
>  
>   s = Size(binaryNumber, /Dimensions)  
>   bn = Reform(Long(binaryNumber), 8*s[1])  
>   len = N_Elements(bn)  
>  
>   RETURN, Total(bn*2^Reverse(Indgen(len)))  
>  
> END  
>  
> Works for the two or three values I've tested. :-)
```

Well, of course it doesn't work for the example *you*
gave, but it works for all the examples *I* used. :-(

Cheers,

David

P.S. Let's just say I *always* prefer long integers to
floats. :-(

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: inverse function of binary.pro
Posted by [eoraptor](#) on Thu, 01 Jul 2004 03:49:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning <davidf@dfanning.com> wrote in message
news:<MPG.1b4c7bbaf53aac1a9897bf@news.frii.com>...

```
> David Fanning writes:
>
>> For something quick and dirty, how about this:
>>
>> FUNCTION Inverse_Binary, binaryNumber
>>
>>   s = Size(binaryNumber, /Dimensions)
>>   bn = Reform(Long(binaryNumber), 8*s[1])
>>   len = N_Elements(bn)
>>
>>   RETURN, Total(bn*2^Reverse(Indgen(len)))
>>
>> END
>>
>> Works for the two or three values I've tested. :-)
>
> Well, of course it doesn't work for the example *you*
> gave, but it works for all the examples *I* used. :-(
>
> Cheers,
>
> David
>
> P.S. Let's just say I *always* prefer long integers to
> floats. :-(
```

How about this one.

If you have an array of binary values called binArr...

```
powers=Lindgen(N_Elements(binArr))
answer=Floor(Total(binArr*2L^(powers),/Double))
```

The assumption above is the binary array prepresents powers of 2 with respect to it's array subscript. Thus binArr[0] is the power of 2^0, binArr[1] is 2^1, etc
