Subject: Inverting indices? Posted by kkobayas on Wed, 29 Jun 1994 16:23:29 GMT View Forum Message <> Reply to Message

Hi,

I guess I didn't say it right in the subject, but what I want to do is, say, for i=[1,3,5] as indeces for a 10-element array, I need to find the indeces for points NOT set by i, that is, j=[0,2,4,6,7,8,9].

Am I making sense? I just need to do j=OPPOSITE(i,10) so that when i=[1,3,5] it'll return j=[0,2,4,6,7,8,9].

I couldn't find a built-in routine that does this. I tried the following code:

FUNCTION opposite, idx, n k = bytarr(n)k(idx)=1return, where (not k) end

but this didn't seem to work for some reason. The shortest working routine I could come up with is:

FUNCTION opposite, index, n k = bytarr(n)k(*)=1k(index)=0 return, where (k) end

I somehow feel there must be a more efficient way of doing this. Can anyone suggest a better solution? Also, why doesn't the first routine work?

| "Would you tell me, please, which way I Ken Kobayashi | ought to go from here?" | "That depends a good deal on where you kkobayas@husc.harvard.edu | want to get to." - Lewis Carroll Subject: Re: Inverting indices?

Posted by robijn on Wed, 29 Jun 1994 20:26:00 GMT

View Forum Message <> Reply to Message

```
In article <kkobayas.772907009@husc9.harvard.edu>,
Ken Kobayashi <kkobayas@husc9.harvard.edu> wrote:
> Am I making sense? I just need to do j=OPPOSITE(i,10) so that when
> i=[1,3,5] it'll return j=[0,2,4,6,7,8,9].
> I couldn't find a built-in routine that does this. I tried the following
> code:
>
> FUNCTION opposite,idx,n
> k = bytarr(n)
> k(idx)=1
> return, where (not k)
> end
>
> but this didn't seem to work for some reason.
A 'false' value in IDL is -1, not 1. So change k(idx)=1 to k(idx)=-1 and
it will work.
   Frank
     / /
             Frank Robijn
                             Internet: Robijn@Strw.LeidenUniv.NL
                                     Bitnet: Robijn@HLERUL51
               Sterrewacht Leiden
          Phone (31) 71 275841
                                   Local: Robijn@HL628
          Fax: (31) 71 275819
                                  Snail: P.O.Box 9513, 2300 RA Leiden,
                              The Netherlands
Subject: Re: Inverting indices?
Posted by dball on Thu, 07 Jul 1994 15:57:28 GMT
View Forum Message <> Reply to Message
In article <2uslco$m4k@highway.LeidenUniv.nl>, robijn@Strw.LeidenUniv.NL (Frank Robijn)
> In article <kkobayas.772907009@husc9.harvard.edu>,
```

- > Ken Kobayashi <kkobayas@husc9.harvard.edu> wrote:

>>

- >> Am I making sense? I just need to do j=OPPOSITE(i,10) so that when
- >> i=[1,3,5] it'll return j=[0,2,4,6,7,8,9].

>>

- >> I couldn't find a built-in routine that does this. I tried the following
- >> code:

```
>>
>> FUNCTION opposite,idx,n
>> k = bytarr(n)
>> k(idx)=1
>> return, where (not k)
>> end
>>
>> but this didn't seem to work for some reason.
> A 'false' value in IDL is -1, not 1. So change k(idx)=1 to k(idx)=-1 and
> it will work.
     Frank
>
```

Well, to be correct, a 'true' value is any 'odd' value, i.e. LSB is 1. A 'false' value is any 'even' value, i.e. LSB is 0. The NOT operator does a bitwise NOT, so this works out. WHERE, however, returns indices of all *nonzero* elements, so use -1 because (NOT -1) is 0.

-- Dave

Subject: Language Documentation; was: Re: Inverting indices? Posted by caron on Fri, 08 Jul 1994 15:49:32 GMT View Forum Message <> Reply to Message

- > Well, to be correct, a 'true' value is any 'odd' value, i.e. LSB is 1.
- > A 'false' value is any 'even' value, i.e. LSB is 0. The NOT operator
- > does a bitwise NOT, so this works out. WHERE, however, returns indices
- > of all *nonzero* elements, so use -1 because (NOT -1) is 0.

Ive been wondering about this. Could RSI please get this kind of stuff into their documents? As a C programmer, I've been used to 0 or not 0 logic. Also, you could mention that all your logical operations are bitwise (I think), e.g "and" is "&" not "&&".

As a general comment to RSI, your manuals seem written for the casual, scientist-type (i.e. non-programmer). Sort of a "dont confuse them" attitude. Your language description is woefully vague to the eyes (ears?) of this programmer. How about a "programmer's description" of the language?

Comments?