Subject: Re: where help
Posted by Phil Williams on Mon, 27 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

DUH!!!!

I should use 256 here and not 255! Became instantly obvious once the
send button was pushed and I admitted my stupidity!!! :)

Take care,
Phil
--
/** ********/
Phil Williams, Ph.D.
Research Instructor
Children's Hospital Medical Center "One man gathers what
Imaging Research Center another man spills..."
3333 Burnet Ave. -The Grateful Dead
Cincinnati, OH 45229
email: williams@irc.chmcc.org
URL: http://scuttle.chmcc.org/~williams/
/** ********/

Subject: Re: where help
Posted by pit on Tue, 28 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <32ED447F.1CFB@irc.chmcc.org>,
Phil Williams <williams@irc.chmcc.org> writes:
> How do I get the x,y coords from the result of where?

Maybe you find the following routine usefull. It should do what you
want for arrays of any dimension. Read the header how to use and
interpret the result.

Peter

FUNCTION Prod, data

on_error, 2

produkt = 1.*data(0)
FOR i=1, n_elements(data)-1 DO $
produkt = produkt*data(i)
return, produkt

```
END

FUNCTION Where_n, data, cond, count
;+
; NAME:
;       WHERE_N
; PURPOSE:
;       Find the n-dim indices where the array DATA fullfills a given
;       condition.
; CALLING SEQUENCE:
;       RESULT= WHERE_N (Array condition, [,COUNT]])
; INPUTS:
;       DATA  : n-dim array expression as explained in the Manpage for
;               where. All Constructs (like "eq 0", "GT limit" etc)
;               are allowed.
; OPTIONAL INPUTS:
;       COUNT : (Output) used for passing back the number of matches
; OUTPUTS:
;       Result is a long array of dimension (COUNT, N) where COUNT is
;       the number of zero elements in DATA and N is the dimension of
;       the input array. The optional parameter COUNT holds the
;       number of matches
; PROCEDURE:
;       An Array expression is a byta array that is unity where the
;       condition is fulfilled and zero elswhere. The where-function
;       returns a 1-d array. Reformat this and return the reformated
;       array.
; EXAMPLE:
;       Be A an 4-dim array. The call
;         Res = where_n(abs(A) LE 10)
;       returns an array of the size (n_matches, 4). For example the
;       4th match element in A can be addressed as
;        A(res(3,0), res(3,1), res(3,2), res(3,3)).
; MODIFICATION HISTORY:
;       20-Okt-1992  P.Suetterlin, KIS 2-d version Where2
;       29-Aug-1995  PS extended to n-dim, changed syntax to match the
;                    use of the IDL where-function.
;-

on_error, 2

IF n_params() EQ 0 THEN BEGIN
   print, 'Use: result=where_n(data[,count])  data is n-dim array'
   return, undefined
ENDIF

s = size(data)
dim = s(0)
```

```
ix = where(data EQ 1, count)

IF count EQ 0 OR s(0) EQ 1 THEN return, ix

res = intarr(count, dim)
res(*, 0) = ix MOD s(1)
FOR i=1, dim-1 DO $
  res(*, i) = (ix MOD long(prod(s(1:i+1))))/prod(s(1:i))

return, res

END
```

--
~~~~~~
Peter "Pit" Suetterlin http://www.uni-sw.gwdg.de/~pit
Universitaets-Sternwarte Goettingen
Tel.: +49 551 39-5048 pit@uni-sw.gwdg.de
-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * --
Come and see the stars! http://www.kis.uni-freiburg.de/~ps/SFB
Sternfreunde Breisgau e.V. Tel.: +49 7641 3492

Subject: Re: where help
Posted by davidf on Wed, 29 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Gunter <gunter@alpha1.csd.uwm.edu> writes:

> David Fanning (davidf@dfanning.com) wrote:
>
> [bunch of nonsense deleted...] :-)

> Is there a reason you wouldn't just write: row=index/s(1)? In this manner you
> will end up with the whole part automatically.
>
> And you could just as well write: col = index MOD s(1), where the MOD function
> returns the remainder of the division (index/s(1)). Of course there may be
> extra time involved (looking up the MOD function, etc) versus the above line.

Yeah, yeah, yeah. I realized all this the minute I hit the SEND key!
It's just that good ideas don't always occur to me when I am
writing programs on the fly. My method of working, to tell you
the truth, is to write some nonsense here, wait around for all
the experts to post or e-mail me the *real* answers, then write
~~~~~~

it up as a tip on my web page like I know what I'm doing.

I figure this is about 100 times faster than looking at the
IDL documentation! :-)

Cheers!

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: where help
Posted by gunter on Wed, 29 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning (davidf@dfanning.com) wrote:

[snip...snip]

```
:    data = FINDGEN(12)
:    data = REFORM(data, 3, 4)
:    PRINT, data
:       0.00000      1.00000      2.00000
:       3.00000      4.00000      5.00000
:       6.00000      7.00000      8.00000
:       9.00000      10.0000      11.0000
:    s = SIZE(data)
:
: OK, suppose I now do this:
:
:    index = WHERE(data EQ 6)
:
: The WHERE function returns the 1D index into the 2D array.
:
:    PRINT, index
:       6
:
: So, the number 6 is located in index 6 (the 7th number in
: the array with zero-based subscripting). What is its
: 2D subscript? Well, if I divide index by how many columns
```

: there are in the array, and then take the whole part of that
: number, I will know its row number. In IDL terms:
:
:   row = FIX(FLOAT(index)/s(1))
:   PRINT, row
:   2

Is there a reason you wouldn't just write: row=index/s(1)? In this manner you will end up with the whole part automatically.

: To find the column number, I multiply the row number times
: the number of columns in the array, and subtract that value
: from the index. Again, in IDL terms:
:
:   col = index - (row * s(1))
:   PRINT, col
:   0

And you could just as well write: col = index MOD s(1), where the MOD function returns the remainder of the division (index/s(1)). Of course there may be extra time involved (looking up the MOD function, etc) versus the above line.

--
david gunter
http://www.mcs.anl.gov/people/gunter/

"When you are a Bear of Very Little Brain, and you Think of Things, you find sometimes that a Thing which seemed very Thingish inside you is quite different when it gets out into the open and has other people looking at it."
- A.A. Milne, "The House At Pooh Corner"

Subject: Re: where help
Posted by fttji on Wed, 29 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Phil Williams <williams@irc.chmcc.org> wrote:

> Here's what I did:
> IDL> t = where(slice eq max(slice))
> IDL> print,size(slice)
>   2   256   256   2   65536
> IDL> y = t/255
> IDL> x = t - (y*255)
Ya, you're using the wrong dimension, use 256
and I prefer x=t mod 256, though I guess yours will work.

> IDL> print,slice(y,x)
You obviously don't really want this in there!

> What is going on??? Any help would be GREATLY appreciated!
Hope this helps!
> Phil
Thomas Immel
Research Asst.
Geophysical Institute, U of Alaska Fairbanks
