Subject: Re: Multiplication turning array into scalar -- who wants to try?
Posted by swingnut on Sun, 01 Apr 2007 04:24:44 GMT
View Forum Message <> Reply to Message

On Apr 1, 12:01 am, David Fanning <n...@dfanning.com> wrote:
...
> Your "signOfDifference" is a one-element array. When you multiply
> an array by another array, the result has as many elements as
> the *smallest* array.
>
> 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.")

Hooray for cross-posting! That's good to know. I guess now I'm just confused, because I've not encountered such a difference in the treatment of scalars and 1x1 arrays. This implies that a scalar is not an array to me despite the spiel about IDL being an array language, so that somewhere in its innards IDL treats a scalar and 1x1 array differently but doesn't always make the result apparent to the user unless said code monkey traces whatever went awry back to this difference of treatment.

Subject: Re: Multiplication turning array into scalar -- who wants to try?
Posted by David Fanning on Sun, 01 Apr 2007 05:01:42 GMT
View Forum Message <> Reply to Message

swingnut@gmail.com writes:

> JOURNALING AT IDL PROMPT:
>
> temp=[0.0,-1.0,0.0]
> print,size(temp)
> ; 1 3 4 3
> temp2=-25.0
> print,size(temp2)
> ; 0 4 1
> print,temp2/ABS(temp2)
> ; -1.00000
> temp2=temp2/ABS(temp2)
> print,SIZE(temp2)
> ; 0 4 1

> print,temp2*temp
> ; 0.000000 1.00000 0.000000
> temp=temp2*temp
> print,temp
> ; 0.000000 1.00000 0.000000
> print,SIZE(temp)
> ; 1 3 4 3
>
> OUTPUT OF SEQUENCE OF PRINT STATEMENTS IN MY CODE:
>
> ;----------
> ;Adjusting density power law in increments of [0.000000 , -1.00000 ,
> 0.000000]
> ;adjustmentIncrement =
> ; 0.000000 -1.00000 0.000000
> ;SIZE of adjustmentIncrement =
> ; 1 3 4 3
> ;-----
> ;deltaVirtHeight =
> ; -0.0678234
> ;SIZE of deltaVirtHeight =
> ; 1 1 4 1
> ;-----
> ;signOfDifference = deltaVirtHeight/ABS(deltaVirtHeight) =
> ; -1.00000
> ;SIZE of signOfDifference =
> ; 1 1 4 1
> ;-----
> ;Multiply signOfDifference with each element of adjustmentIncrement,
> i.e.,
> ;adjustmentIncrement[i]=adjustmentIncrement[i]*signOfDiffere nce,
> i=0,1,2:
> ;adjustmentIncrement =
> ; 0.000000 1.00000 0.000000
> ;SIZE of adjustmentIncrement =
> ; 1 3 4 3
> ;-----
> ;But if we do
> adjustmentIncrement=adjustmentIncrement*signOfDifference, it becomes a
> scalar?
> ;adjustmentIncrement =
> ; 0.000000
> ;SIZE of adjustmentIncrement =
> ; 1 1 4 1
> ; % Attempt to subscript LOCDENSPL with <INT (1)> is out
> of range.

It is a little hard to discern a question here, but

I think you are confusing yourself by using the SIZE
function. Using a HELP command might help more. Here
is what is happening:

```
IDL> a = [1,2,3]
IDL> b = [3]
IDL> print, a*b
       3
IDL> print, size(b)
          1           1           2           1
IDL> help, b
B               INT       = Array[1]
IDL> b = 3
IDL> print, a*b
       3       6       9
IDL> help, b
B               INT       =        3
IDL> print, size(b)
          0           2           1
```

Your "signOfDifference" is a one-element array. When you multiply
an array by another array, the result has as many elements as
the *smallest* array.

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: Multiplication turning array into scalar -- who wants to try?
Posted by David Fanning on Sun, 01 Apr 2007 07:13:48 GMT
View Forum Message <> Reply to Message

swingnut@gmail.com writes:

> Hooray for cross-posting! That's good to know. I guess now I'm just
> confused, because I've not encountered such a difference in the
> treatment of scalars and 1x1 arrays. This implies that a scalar is not
> an array to me despite the spiel about IDL being an array language, so
> that somewhere in its innards IDL treats a scalar and 1x1 array
> differently but doesn't always make the result apparent to the user
> unless said code monkey traces whatever went awry back to this
> difference of treatment.

Oh, oh. Don't even get started along this path, or it's
all we are going to talk about all week. IDL just is what
it is. :-(

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.")
