Subject: Re: Array subscripting oddity
Posted by jdlb on Thu, 19 Aug 1993 01:25:28 GMT
View Forum Message <> Reply to Message

glenn@atm.ch.cam.ac.uk (Glenn Carver) writes:
[Sample array indexing problem, showing strange behavior:]

```
glenn> WAVE> a=indgen(3)
glenn> WAVE> m=[45,99]
glenn> WAVE> a(m)=255
glenn> WAVE> print,a
glenn>      0      1    255
```

glenn> Now, I would have thought that because both elements of 'm' are more
glenn> than the size of 'a', no element of 'a' would have been assigned?

kevin@dipl.rdd.lmsc.lockheed.com (Kevin Anderson) responds:
kevin> Looks like a bug to me.

I don't know about PV-WAVE, but in IDL it's not really a bug--it's a strange
but documented feature. From the IDL v. 3.0 manual:

Page 5-5, in "Array Subscripts" section:

> If an element of the subscript array [M in the example above] is
> greater than or equal to the last subscript in the subscripted
> variable [A, above], the last element [of A] is selected.

Similarly, a negative subscript refers to the first (0th) element of the
array. The same information is expressed in different words on p. 6-4, in
subsection "Using Array Subscripts with the Second Form of the Assignment
Statement."

It would be more logical, and easier to avoid errors, if this "feature" were
not present. After all, a scalar subscript causes an error if out of bounds.
And a statement like

```
  b(where(b EQ c)) = d
```

makes b(0)=d if there are no elements where b=c, because the WHERE statement
returns -1 if no elements are found.

glenn> The values '45' and '99' are completely arbitrary. You can set them to
glenn> anything you like.

kevin> the values of m are not quite arbitrary. It works right if
kevin> these values are in the proper range for the size of a.

Like Glenn, I was able to reproduce this behavior with arbitrary values of M.

--Jeff

% J-F de La Beaujardiere % jdlb@ifa.hawaii.edu %
% Institute for Astronomy %   808-956-9843   %
% University of Hawai`i %   fax 956-9402   %