
Subject: Why doesn't this return the correct value?

Posted by [chloesharrocks](#) on Thu, 14 Feb 2008 10:59:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear all

I currently have an array called `good_index`. Some of the elements are ≥ 0 and others are < 0 . I want to write a code that firstly finds how many of these elements are greater than zero and use this number to produce another array (called `good_indexed_precip`) with that number of elements. Then I want it to search through each element of `good_index` in turn. If upon searching it finds that the element is ≥ 0 I want it to find the value of that element in `good_index`. I then want to find the data stored in the index given by that value in another array called `precip_change`. The code I've written is below:

```
=====
counter_2 = 0
number = total(good_index ge 0, /int) ;this counts how many elements
in good_index are  $\geq$  zero
print, number ;this gives the correct answer 6
```

```
FOR s=0, (N_ELEMENTS(good_index)-1) DO BEGIN
  good_indexed_precip=fltarr(number)
  IF good_index[s] ge 0 THEN BEGIN
    good_indexed_precip[counter_2] = precip_change[good_index[s]]
    counter_2++
  ENDIF
ENDFOR
=====
```

Unfortunately, it then prints `good_indexed_precip` with all 6 elements zero, even though the values of `precip_change[good_index[s]]` are non-zero.

If I were to put in the data by hand, eg say
`good_indexed_precip[0] = precip_change[good_index[0]]`
`good_indexed_precip[1] = precip_change[good_index[13]]`
This works fine, so I can't see why my loop isn't working correctly. I know that the 0th & 1st element of `good_index` are both $\geq$ zero, yet if I run the loop: `FOR s=0, 1 DO BEGIN` etc and print `good_indexed_precip` it only has a non-zero value in the 1st element and not the 0th element!

Thanks for all your help in advance.
Chloé
