Subject: Re: Counting the lengths of continuous numbers in an array
Posted by mankoff on Fri, 08 Aug 2008 21:28:20 GMT
View Forum Message <> Reply to Message

On Aug 8, 5:05 pm, andy.mcn...@gmail.com wrote:
> Howdy folks.
>
> Say I have the following array: [1, 2, 3, 4, 7, 8, 11, 13, 14, 17,
> 18, 19]
>
> How would I tell IDL to go through this array and return the array [4,
> 2, 1, 2, 3].....the lengths of the continuous groups of numbers in the
> array?
>
> Thanks!
> -Andy McNeel

I'm not sure how robust this is for other arrays, but I can get your
2nd array from your first like so, assuming the 1st array is "a". I
didn't test boundary cases (runlength of 1 on beginning or end) or
much else...

```
IDL> aa = [a[0],a,0]
IDL> b = where( (aa-shift(aa,1)) ne 1)
IDL> c = b-shift(b,1)
IDL> print, c[1:*]
       4       2       1       2       3
```

 -k.

Subject: Re: Counting the lengths of continuous numbers in an array
Posted by andy.mcneel on Fri, 08 Aug 2008 21:41:02 GMT
View Forum Message <> Reply to Message

On Aug 8, 4:28 pm, mankoff <mank...@gmail.com> wrote:
> On Aug 8, 5:05 pm, andy.mcn...@gmail.com wrote:
>
>> Howdy folks.
>
>> Say I have the following array: [1, 2, 3, 4, 7, 8, 11, 13, 14, 17,
>> 18, 19]
>
>> How would I tell IDL to go through this array and return the array [4,
>> 2, 1, 2, 3].....the lengths of the continuous groups of numbers in the
>> array?
>
>> Thanks!

>> -Andy McNeel
>
> I'm not sure how robust this is for other arrays, but I can get your
> 2nd array from your first like so, assuming the 1st array is "a". I
> didn't test boundary cases (runlength of 1 on beginning or end) or
> much else...
>
> IDL> aa = [a[0],a,0]
> IDL> b = where((aa-shift(aa,1)) ne 1)
> IDL> c = b-shift(b,1)
> IDL> print, c[1:*]
>   4   2   1   2   3
>
>   -k.

This worked beautifully. Thanks! -Andy

Subject: Re: Counting the lengths of continuous numbers in an array
Posted by Brian Larsen on Mon, 11 Aug 2008 13:08:45 GMT
View Forum Message <> Reply to Message

>> IDL> aa = [a[0],a,0]
>> IDL> b = where((aa-shift(aa,1)) ne 1)
>> IDL> c = b-shift(b,1)
>> IDL> print, c[1:*]
>>   4   2   1   2   3

While I can't seem to find it this second my tired and jet lagged
brain is telling me that there is an built-in idl function to
calculate array[*]-array[1:*], which should be equivalent to above.
Now to remember the function.

Cheers,

Brian