Subject: Re: Getting rid of loops
Posted by Michael Asten on Fri, 18 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Try this.
Good luck,
Michael Asten

```
 function fn,cycles
   cycles_and=[cycles,max(cycles)]
   cycles_shift=[cycles[0],cycles]
     print , cycles_and
     print , cycles_shift
   b=where(cycles gt cycles_shift)
   sub_cycles=[ cycles_shift[b[0]], cycles_and[b] ]
   return,sub_cycles
 end


 cycles=[2,2,2,2,2,6,6,7,7,7,9,9,9,27,27,27]
 print,fn(cycles)
;[2,6,7,9,27]>
 end
```

Jacob Noel-Storr wrote:

> I am trying to create non-loopy IDL code for a project I am working on.
>
> I have a very long list of integers ('cycle numbers') and I need some way of
> creating an array that consists of just the cycle numbers. For example if I
> had:
>
>> cycles=[2,2,2,2,2,6,6,7,7,7,9,9,9,27,27,27]
>
> then I would want a function which would do this kid of thing:
>
>> print,fn(cycles)
> [2,6,7,9,27]
>
> At the moment this is done by looping through the data twice. Once to count
> how many different values there are and once to write the different values
> into an array.
>
> Any suggestions about how this could be achieved without the two 'for'
> loops?
>
> Thanks,

> Jacob
>
> --
> Jacob Noel-Storr
> Astronomy Department
> Columbia University / Biosphere 2 Center
> www.astro.bio2.edu/jacob/
> jake@astro.columbia.edu