Subject: Re: Unpacking algorithm
Posted by MKatz843 on Mon, 17 May 2004 04:37:17 GMT

View Forum Message <> Reply to Message

You could also use where() to find the indices of the "zero" elements,
or their complement, that is the indices of the non-zero elements.

```
v0=[0 x1 x2 x3 0 x4 x5 x1 0 x2 x3 x4 0 x5 x1 x2 0 x3 x4 x5]
w2 = where(v0 NE 0, complement=w1)  ;--- returns an array of indices
v2 = v0(w2)
v1 = v0(w1)
```

There are also very handy keywords: count and ncomplement that return
the number of indices of each kind that are found. Note that where()
returns -1 when there are no matches. If you try to use v0(w1) and w1
is -1 you'll get an error. That's where the count variables come in. I
usually do something like "if count GT 0 then (do something)" to avoid
the error.

M. Katz

"Tmorri" <torrimorri@yahoo.com> wrote in message
news:<52e0072063f360e0705e8ac96f274e52@localhost.talkaboutprogramming.com>...

> one second thought,
>
> Does any one have an algorithm to unpack this vector
>
>
> v0=[0 x1 x2 x3 0 x4 x5 x1 0 x2 x3 x4 0 x5 x1 x2 0 x3 x4 x5]
>
> in the following way:
>
> v1=[0 0 0 0 0]
>
> v2=[x1 x2 x3 x4 x5 x1 x2 x3 x4 x5 x1 x2 x3 x4 x5]
>
> x1,x2,x3,x4,x5 are variables that can take any value, even zero, (0).
>
> I just want to get rid othe zeroes (every fourth element)shown in vector
> v0
>
>
> Thanks,
>
> Tmorri