Subject: Re: Unpacking algoritmn
Posted by David Fanning on Sat, 15 May 2004 16:12:37 GMT
View Forum Message <> Reply to Message

Tmorri writes:

> Does any one have an IDL algorithm to unpack this vector
>
> v0=[0 1 2 3 0 4 5 1 0 2 3 4 0 5 1 2 0 3 4 5]
>
> in the following way:
>
> v1=[0 0 0 0 0]
>
> v2=[1 2 3 4 5 1 2 3 4 5 1 2 3 4 5]

Do you mean something like this:

```
IDL> print, v0
 0 1 2 3 0 4 5 1 0 2 3 4 0 5 1 2 0 3 4 5
IDL> v1 = v0[where(v0 eq 0)]
IDL> print, v1
 0 0 0 0 0
IDL> v2 = v0[where(v0 ne 0)]
IDL> print
 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Cheers,

David
--

David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Unpacking algoritmn
Posted by Tmorri on Sat, 15 May 2004 23:03:46 GMT
View Forum Message <> Reply to Message

Thanks for your replay. I made a little mistake in the statement of the
problem, what I really need is this:

x1,x2,x3,x4,x5 are variables that can take any value, even zero, (0).

I just want to get rid othe zeroes shown in vector v0 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]

Thanks for your help.

Tmorri

Subject: Re: Unpacking algoritmn
Posted by James Kuyper on Mon, 17 May 2004 14:40:21 GMT
View Forum Message <> Reply to Message

Tmorri wrote:
> Thanks for your replay. I made a little mistake in the statement of the
> problem, what I really need is this:
>
> x1,x2,x3,x4,x5 are variables that can take any value, even zero, (0).
>
> I just want to get rid othe zeroes shown in vector v0
>
> 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]

Examples can be useful for illustrating a problem definition, but
they're a poor method for actually defining the problem, as you've
already seen.

The key issue here, is how should the code you're looking for identify
which values to put into v1, and which ones to put into v2? David
Fanning's answer was based upon one assumption, which was compatible
with your first example: that the items to go into v0 are the ones with
a value of 0. Now we know that it's what you want, we have to make a
different guess.

My best guess is that you want to put into v1 every fourth element,
starting with element 0. You want the rest of the elements in v2. Is
that correct? Will the total number of elements always be a multiple of
4, or will there sometimes be a few left over? Will the total number of

elements always be exactly 20?

Subject: Re: Unpacking algoritmn
Posted by tam on Mon, 17 May 2004 16:04:36 GMT

View Forum Message <> Reply to Message

Tmorri wrote:

> Thanks for your replay. I made a little mistake in the statement of the
> problem, what I really need is this:
>
> x1,x2,x3,x4,x5 are variables that can take any value, even zero, (0).
>
> I just want to get rid othe zeroes shown in vector v0
>
> 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]
>
> Thanks for your help.
>
> Tmorri
>
>

Then something like:

```
  v2 = v0[ (lindgen(n_elements(v0)) mod 4) ne 0]
```

might do the trick. In practice I'd usually split
it out into a couple of lines.

```
  index = lindgen(n_elements(v0))
  v2    = v0[ (index mod 4) ne 0]
```

Regards,
Tom McGlynn
