Subject: Re: WHERE Function
Posted by chase on Thu, 11 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Pryhoda" == PRYHODA <bjp8350@osfmail.isc.rit.edu> writes:
In article <1995May10.175428.15046@ultb.isc.rit.edu> bjp8350@osfmail.isc.rit.edu (PRYHODA) writes:

Pryhoda> How can I seperate the LONGWORD VECTOR into its X and Y
Pryhoda> components? so I can use the distance formula. Or is there an
Pryhoda> easier way to find the distance between each pixel?

I have a function the changes a one-dimensional index for an array
into its multi-dimensional version. It more generally performs
decoding for arbitrary dimensions/radix/bases.

It is not necessarily the best solution for your particular problem,
but I find it useful in a large variety of instances.

I have included the function below. See the EXAMPLE section in the
header.

I hope this is useful. If you have improvements, please let me know.

Thanks,
Chris

--- Begin included file ------
```
FUNCTION Decode, scl, dim, help=help, quiet=quiet
;+
; $Id: decode.pro,v 1.4 1995/05/10 16:20:27 chase Exp $
;
; NAME:
;
;   DECODE
;
; PURPOSE:
;
;   Decode a vector of scalars into the dimensions d1,...,dn in order
;   of increasing significance.  Useful for conversions between
;   different base/radix, time conversions, etc.  See EXAMPLE below.
;
; CATEGORY:
;
;   Mathematical Functions
;
; CALLING SEQUENCE:
;
```

```
;   Result = DECODE(Scl, Dim)
;
; INPUTS:
;
;   Scl - Vector of scalars to be decoded.
;
;   Dim - If a scalar, then it is used as a repeated base for the
;         decoding, i.e., D1,...,DN=Dim.
;         If a 1 dimensional vector, then it is taken as the
;         dimensions D1,...,DN.
;         If > 1 dimensional array, then the dimensions of the array
;         are used for D1,...,DN.
;         The dimensions increase in significance, i.e., the first
;         dimension is the least significant and the last dimension is
;         the most significant.
;
;         Dim need not be integral.
;
;         D1*D2*...*DN must be representable to the precision of a double
;         for the results to be accurate.
;
; KEYWORD PARAMETERS:
;
;   HELP - Provide help (this information). No other action is performed.
;
;   QUIET - Do not print warning when Scl is outside total dimensions.
;
; OUTPUTS:
;
;   Result - Array of size NxM where M is dimension of Scl.  Array has
;            the same type as Scl.  Result(*,i) is the decoding of the
;            scalar Scl(i). If Scl(i) is larger then the dimensioned
;            size, i.e. D1*D2*...*DN, then the modulus, Scl(i) mod
;            D1*D2*...*DN, is decoded.  Result(j-1,i) corresponds to
;            dimension Dj with Result(N-1,i) the most significant
;            digit of the decoding.
;
; PROCEDURE:
;
;   Let b0,...,bN be the decoding. Then Scl can be represented as:
;
;   Scl = D1*D2*...*D[N-1]*bN + ... + D1*D2*b3 + D1*b2 + b1
;
;       = D1(D2(...(D[N-1]*bN + b[N-1])...+ b2 ) + b2) + b1
;
;
;
;   with 0 <= bi < Di, and b2,...,bN integral.
;   If Scl is floating point then b1 may not be integral.
```

```
;
;   The representation is unique for Scl, i.e., there are not two
;   distinct decodings resulting in the same Scl.
;
; EXAMPLE:
;
;   scl = [20,63]
;   ; Conversion to base 16
;   print,decode(scl,16)
;
;   ; Convert times in seconds to second, minute, hour, day
;   t = [1.,60.,3600.d0,24.*3600d0] # [[32,55,10,2],[59,0,23,364]]
;   print,t
;   print,decode(t,[60,60,24,365])
;
;   ; Conversion to binary (base 2)
;   print,decode(scl,2)
;   ; Invert the decoding
;   print,2^indgen(5)#decode(scl,2)
;
;   ; Arbitrary decoding.  Generates a warning for decoding 63 in
;   ; which case (63 mod 3*4*5) = 3 is decoded.
;   print,decode(scl,[3,4,5])
;   print,[1,3,3*4]#decode(scl,[3,4,5])
;   print,[1,3,3*4,3*4*5]#decode(scl,[3,4,5,6])
;
;   ; Convert 1D index into a multi-dimensional index
;   w=dist(20,20)
;   a=max(w,i)
;   ; Get 2D index for max
;   print,decode(i,w)
;
; MODIFICATION HISTORY:
;
;       Mon Feb 27 16:13:20 1995, Chris Chase S1A
;       <chase@retro.jhuapl.edu>
;
;               Handles non-integral dimensions and inputs.
;
;       Mon Jul 18 15:58:18 1994, Chris Chase S1A <chase@jackson>
;               Fixed/cleaned up.
;
;       Mon Jul 26 12:17:56 1993, Chris Chase <chase@aphill>
;               Created.  Named for similar APL function.
;
;-
if keyword_set(help) then begin
   doc_library, 'decode'
```

```
    return, ''
endif
s = size(dim)
if (s(0) eq 0) then begin
    d = replicate(dim, ceil(alog(max(scl))/alog(dim)))
endif else begin
    if (s(0) gt 1) then d = s(1:s(0)) $
    else d = dim
endelse

nd = n_elements(d)

;; Use double to make sure it is big enough
dd = double(d)
for i=1, nd-1 do dd(i) = dd(i)*dd(i-1)
v = scl
if max(v/dd(nd-1)) gt 1 then begin
    if not keyword_set(quiet) then begin
        print, 'Warning - function DECODE: scalar outside dimension ' + $
          'bounds, decode of modulus returned.'
    endif
    v(*) = v(*) mod dd(nd-1)
endif

index = replicate(v(0), nd, n_elements(v))
for i = nd-1, 1, -1 do begin
    f = long(v/dd(i-1))
    index(i, *) = f
    v = v-f*dd(i-1)
endfor
index(0, *) = v
return, index
end
```


--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu