Subject: Re: converting int array to a string
Posted by Richard French on Wed, 21 Nov 2001 01:11:04 GMT
View Forum Message <> Reply to Message

Chad Bahrmann wrote:
>
> Is there an easy way to convert the following:
>
> x=[1,2,3,4,5,17,45,46,47,48,49,50,51,59]
> where x is minutes in an hour for example
>
> to a string like:
>
> minutes='1-5,17,45-51,59'
>
> Thanks in advance,
>
> Chad Bahrmann

Chad - Not that I know of, but I think I have hacked something
together before using the shift operator to tell you which elements
are adjacent. For example:

```
x=[1,2,3,4,5,17,45,46,47,48,49,50,51,59]

dx=(x-shift(x,1))[1:*]
print,dx
1    1    1    1   12   28    1    1    1    1    1    1
8
; then you can figure out which ones require special attention:

l=where(dx ne 1)
; the following prints the last element in a seqeuence that can be
; connected by a hyphen
print,x[l]

; but I think that handling all the special cases could be a nightmare!
```

Post your best solution! This is offered as just a start.
Dick French

Subject: Re: converting int array to a string
Posted by Chad Bahrmann on Wed, 21 Nov 2001 05:32:06 GMT
View Forum Message <> Reply to Message

Thanks for the headstart...I believe I have something that works....
The function I have written below gives:
IDL> x=[1,2,3,5,6,7,9,45,46,47,48,49,50,51,56,59]

```
IDL> print, time2str(x)
% Compiled module: TIME2STR.
1-3,5-7,9,45-51,56,59

function time2str, x
dx=(x-shift(x,1))[1:*]
l=where(dx ne 1)
times=''
nl=n_elements(l)
nx=n_elements(x)
if l[0] eq -1 then times='0-59' else begin
  if l[0] eq 0 then times=strtrim(string(x[0]),2)+','
    if l[0] ne 0 then
 times=strtrim(string(x[0]),2)+'-'+strtrim(string(x[l[0]]),2) +','
    for i=0, nl-2 do begin
      if ((l[i+1]-l[i]) gt 1) then begin
         times=times+strtrim(x[l[i]+1],2)+'-'+strtrim(x[l[i+1]],2)+', '
      endif else begin
        times=times+strtrim(x[l[i+1]],2)+','
      endelse
    endfor
  if (nx-1)-l[(nl-1)] eq 1 then times=times+strtrim(string(x[nx-1]),2) else
$
   times=times+strtrim(x[(l[nl-1])+1],2)+'-'+strtrim(string(x[n x-1]),2)
endelse
return, times
END
```

"Chad Bahrmann" <cbahrmann@ou.edu> wrote in message
news:v8CK7.6670$o4.14684@news.ou.edu...
> Is there an easy way to convert the following:
>
> x=[1,2,3,4,5,17,45,46,47,48,49,50,51,59]
> where x is minutes in an hour for example
>
> to a string like:
>
> minutes='1-5,17,45-51,59'
>
> Thanks in advance,
>
> Chad Bahrmann
>
>

Subject: Re: converting int array to a string

> function time2str, x

Chad -

I took the liberty of adding to your program a little. Notice that if
you put in:

IDL> print, time2str([1,2,3])

you'd get back '0-59'. My revised function below fixes this and adds
support for all ranges (with a min and max keyword so you can set your
time2str function up). It also removes duplicate elements, and sorts
the array. I also got rid of all the calls to strtrim, and replaced
them with one call to strcompress at the end.

Todd

--

```
function time2str, x
  return, getnumberrange( x, min=0, max=59 )
end ;; time2str

function getnumberrange, x, min=min, max=max
  if( n_elements( x ) eq 0 ) then return, ''

  tempX = x
  if( n_elements( min ) gt 0 ) then begin
     good = where(tempX ge min[0], ngood )
     if( ngood gt 0 ) then $
       tempX = tempX[good] $
     else return, ''
  endif

  if( n_elements( max ) gt 0 ) then begin
     good = where(tempX le max[0], ngood )
     if( ngood gt 0 ) then $
       tempX = tempX[good] $
     else return, ''
  endif

  newX = tempX[uniq( tempX, sort(tempX) )]

  ;; No unique elements, just return first element
  if( n_elements( newX ) eq 1 ) then return, strtrim( tempX[0],2 )
```

```
  ;; See which elements in new X differ by only 1 (they will become
  ;; part of a range of numbers)
  dx=(newX-shift(newX,1))[1:*]

  ;; Get number of elements where spread is greater than 1
  ;; (non-contiguous elements)
  l=where(dx ne 1)

  outStr=''
  nl=n_elements(l)
  nx=n_elements(newX)

  if l[0] eq -1 then outStr=string( newX[0],'-',newX[nx-1]) else begin
     if l[0] eq 0 then outStr=string(newX[0])+','
     if l[0] ne 0 then $
       outStr=string(newX[0])+'-'+string(newX[l[0]])+','
     for i=0, nl-2 do begin
        if ((l[i+1]-l[i]) gt 1) then begin
            outStr=outStr+string(newX[l[i]+1])+'-'+string(newX[l[i+1]])+ ','
        endif else begin
           outStr=outStr+string(newX[l[i+1]])+','
        endelse
     endfor
     if (nx-1)-l[(nl-1)] eq 1 then $
       outStr=outStr+string(newX[nx-1]) else $
        outStr=outStr+string(newX[(l[nl-1])+1])+'-'+string(newX[nx-1 ])
  endelse
  return, strcompress(outStr, /remove_all)
end ;; getnumberrange
```