Subject: Re: sorting string arrays - non alphabetic and user defined order
Posted by greg michael on Mon, 16 Oct 2006 12:03:08 GMT

View Forum Message <> Reply to Message

Try this...

```
namelist=['Daddy','Groggy','Ally','Curry','Emmy','Bully','Jo ckey','Hippy','Itchy','Fluffy']
mylist=['Emmy','Fluffy','Itchy','Jockey']
n=n_elements(namelist)
m=n_elements(mylist)
q1=rebin(transpose(indgen(m)),n,m)
q2=rebin(indgen(n),n,m)
s=mylist[q1] eq namelist[q2]
print,namelist[where(total(s,2) gt 0)]
```

regards,
Greg

Subject: Re: sorting string arrays - non alphabetic and user defined order
Posted by rkombiyil on Tue, 17 Oct 2006 04:42:14 GMT

View Forum Message <> Reply to Message

Hi Greg,

Thank you for your quick response, it is much appreciated. Your method works fine. I am afraid I left out of couple of questions related to the problem.

Is there a way to dynamically allocate arrays (varying size) ?
Specifically, this is what I am trying to do.

#1

I have a station database. This "string" array contains details pertaining to all the stations (names,locations,code etc.). Let's call this NAMELIST 'A'

#2

Now, I made a list of stations I want to look at from the above database. Let's call this NAMELIST 'B'

#3

The list of stations provided to me (NAMELIST 'C') may or may not contain all the stations in namelist 'B'

#4

I make a 1-1 string match between namelists 'B' and 'C' and extract
only those stations that are present in B and C, and order C similar to
B (user defined non-alphabetic)

#5

Problem is: I know the dimensions of the string array 'B' because I get
to choose the stations I want from the original big database. Now, I
don't know the dimensions of 'C' because it is variable, it may have
same dimensions as B or less than B or none at all and the order might
vary. Since, it can "ALSO" have same dimensions of B (max possible
dimension), I define

C=strarr(n_elements(B))

But when the dimensions are less than that in B, there are elements
with 'strlen' equal to zero. For example, C[3] maybe empty and may be
C[7] and all other elements of the array may be filled.. I want to get
rid of these trailing/beginning/in between empty (zero length) elements
(I know this is because of the above declaration)
and make this array to have dimensions = # of non-zero elements

Is there a way to tackle such situations? I tried to index and
increment the # of non-zero elements and redeclare the dimensions of C
to be # of non-zero elements.. But it didn't work.. Meanwhile, I will
try to modify your code to see if it works for my need.

I appreciate your time and help!
Thanks much,
~rk