Subject: Help
Posted by psumartoira on Fri, 02 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hi, There.

I am a beginner in IDL. I understand that there is a free software
to analyze the 2D cross-correlation function of two images and find the optimal
(x,y) offset (shifting pixel of one image relative to the other.

Thanks

Irawan
Curtin University,PERTH,AU
d@D-ware

Subject: Re: Help
Posted by Pavel Romashkin on Fri, 21 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

The task is not strictly defined: the two matrices Tri provided are
different in size. Therefore, I treated it as a general problem. Here is
a 1-line solution for any size matrix:

print, (bindgen(k*m+m-1, x) mod m) eq 0

Here,
m is the length of the cycle within a row,
k is an integer multiplier to obtain the necessary row length,
x - arbitrary number of rows.

Example with m=4, k=3, x=5:

```
IDL> print, (bindgen(3*4+3, 5) mod 4) eq 0
   1   0   0   0   1   0   0   0   1   0   0   0   1   0   0
   0   1   0   0   0   1   0   0   0   1   0   0   0   1   0
   0   0   1   0   0   0   1   0   0   0   1   0   0   0   1
   0   0   0   1   0   0   0   1   0   0   0   1   0   0   0
   1   0   0   0   1   0   0   0   1   0   0   0   1   0   0
```

Resulting matrix can be trimmed in any fashion you desire. Or, maybe I
could just type matrices in, if they were needed precisely and only in the
way Tri sent them?...
Cheers,
Pavel

Tri VU KHAC wrote:

> Hi folks,
>
> I'm looking for an efficient way to produce the matrix of type (I woulk
> like to be able to avoid the FOR statement)
>
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
>
> OR
>
> 1 0 0 1 0 0 1
> 0 1 0 0 1 0 0
> 0 0 1 0 0 1 0
> 1 0 0 1 0 0 1
>
> ETC.
>
> Best regards,
> Tri.

Subject: Re: Help
Posted by Martin Schultz on Fri, 21 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Tri VU KHAC wrote:
>
> Hi folks,
>
> I'm looking for an efficient way to produce the matrix of type (I woulk
> like to be able to avoid the FOR statement)
>
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
>
> OR
>
> 1 0 0 1 0 0 1
> 0 1 0 0 1 0 0
> 0 0 1 0 0 1 0
> 1 0 0 1 0 0 1
>
> ETC.
>

> Best regards,
> Tri.

Isn't that coincidence? Take a look at the recent discussion on "AN array slicing function". I posted a routine ARREX that will do the opposite of what you are looking for, i.e. you could extract all the 1's from your matrix with it (probably need several calls though). Now you can go ahead and use the function arrex_ComputeInd (included in arrex.pro) to come up with a function arrexi that returns the indices rather than the values, and then you would write

```
A = intarr(8,4)
A[arrexi(A,[1,0],-1,[2,2]) = 1
A[arrexi(A,[0,1],-1,[2,2]) = 1
```

(or similar) ... If you wait a little, I'll hack arrexi for you as soon as soem other folks have confirmed that arrex works.

Regards,
Martin.

--

|||||||||||||||\\\\\\\\\\\\\\\----------////////////// //|||||||||||||||
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

Subject: Re: Help
Posted by R.Bauer on Tue, 25 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I like to use matrix operations:

```
a=[1, -1, 1, -1, 1, -1, 1, -1]
b=[-1, 1, -1, 1, -1, 1, -1, 1]

res=a#b
print,res gt 0
```

Cia,

R

Tri VU KHAC wrote:

> Hi folks,
>
> I'm looking for an efficient way to produce the matrix of type (I woulk
> like to be able to avoid the FOR statement)
>
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
> 0 1 0 1 0 1 0 1
> 1 0 1 0 1 0 1 0
>
> OR
>
> 1 0 0 1 0 0 1
> 0 1 0 0 1 0 0
> 0 0 1 0 0 1 0
> 1 0 0 1 0 0 1
>
> ETC.
>
> Best regards,
> Tri.

Subject: Re: Help
Posted by Vapuser on Wed, 26 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

"R.Bauer" <R.Bauer@fz-juelich.de> writes:

Very slick!

William

> I like to use matrix operations:
>
> a=[1, -1, 1, -1, 1, -1, 1, -1]
> b=[-1, 1, -1, 1, -1, 1, -1, 1]
>
> res=a#b
> print,res gt 0
>
> Cia,
>
> R
>
>

> Tri VU KHAC wrote:
>
>> Hi folks,
>>
>> I'm looking for an efficient way to produce the matrix of type (I woulk
>> like to be able to avoid the FOR statement)
>>
>> 0 1 0 1 0 1 0 1
>> 1 0 1 0 1 0 1 0
>> 0 1 0 1 0 1 0 1
>> 1 0 1 0 1 0 1 0
>>
>> OR
>>
>> 1 0 0 1 0 0 1
>> 0 1 0 0 1 0 0
>> 0 0 1 0 0 1 0
>> 1 0 0 1 0 0 1
>>
>> ETC.
>>
>> Best regards,
>> Tri.
>

--
William Daffer: 818-354-0161: vapuser@catspaw.jpl.nasa.gov

Subject: Reading ASCII files as structure: Help
Posted by limiqt on Thu, 01 Oct 2015 02:15:07 GMT
View Forum Message <> Reply to Message

Hi everyone,

I was wondering if someone could assist me in reading ascii files as structures. I am followin the example shown in http://www.idlcoyote.com/tips/ascii_column_data.html.

mydata.dat is similar to

456.34 23.982 4.8 9/10/2012, 9/15/2012 Brazil
3491.33 10003.1 10.0 8/10/2014 10/10/2014 USA
333.2 1.3 100.2 5/5/2010 8/20/2010 USA
1211.84 22.4 82.2 10/15/2014 10/20/2014 UK
...

I need to read mydata.dat as var1, var2, var3, datei, datef, cnt

I have done the following:
==========
Pro mydataASCII
close, /all

file = 'mydata.dat'
dataStruct = {var1:0.0, var2:0.0, var3:0.0, datei:' ', datef:' ', cnt:' ' }

nrows = File_Lines(file)
data = Replicate(dataStruct, nrows)

OpenR, lun, file, /GET_LUN
ReadF, lun, data
Free_Lun, lun

End

==========

However the program stop in ReadF, lun, data with the meassages:
% READF: End of file encountered. Unit: 100
Execution halted at: mydataASCII   11

could someone show me what i am doing wrong?.

Thanks

Lim.

Subject: Re: Reading ASCII files as structure: Help
Posted by penteado on Fri, 02 Oct 2015 02:25:50 GMT
View Forum Message <> Reply to Message

On Wednesday, September 30, 2015 at 11:15:18 PM UTC-3, Lim wrote:
> However the program stop in ReadF, lun, data with the meassages:
> % READF: End of file encountered. Unit: 100
> Execution halted at: mydataASCII   11
>
> could someone show me what i am doing wrong?.

It sounds like the problem happens because you have 3 fields that are strings, and readf is reading them without an explicit format. By default, readf will read a string up to the end of the line. If you do not supply a format, it has no way of knowing at which character the fields datei and datef end, so it reads datei to the end of the line, then will read datef from the next line. You can check to see if that is the case by printing the values of your data array, to see how it was filled up by readf up to the point the program stopped (because it hit the end of the file).

You have to either give readf an explicit format, or read the lines, split them and give each field the proper value. If you have a recent version of my library (pp_lib, http://ppenteado.net/idl/pp_lib/doc/index.html), you can read it just with:

data=pp_parsetext('mydata.dat',nheader=0,delimiter=',
',/as_struct,fieldnames=['var1','var2','var3','datei','datef ','cnt'])

When I do that with your example data, I get:


```
IDL> data
[
   {
      "VAR1": 456.33999999999997,
      "VAR2": 23.981999999999999,
      "VAR3": 4.7999999999999998,
      "DATEI": "9/10/2012",
      "DATEF": "9/15/2012",
      "CNT": "Brazil"
   },
   {
      "VAR1": 3491.3299999999999,
      "VAR2": 10003.100000000000,
      "VAR3": 10.000000000000000,
      "DATEI": "8/10/2014",
      "DATEF": "10/10/2014",
      "CNT": "USA"
   },
   {
      "VAR1": 333.19999999999999,
      "VAR2": 1.3000000000000000,
      "VAR3": 100.20000000000000,
      "DATEI": "5/5/2010",
      "DATEF": "8/20/2010",
      "CNT": "USA"
   },
   {
      "VAR1": 1211.8399999999999,
      "VAR2": 22.399999999999999,
      "VAR3": 82.200000000000003,
      "DATEI": "10/15/2014",
      "DATEF": "10/20/2014",
      "CNT": "UK"
   }
]
```