Subject: Swap even/odd elements in array
Posted by photosalex on Wed, 22 Jun 2005 08:16:14 GMT
View Forum Message <> Reply to Message

Hi All,
sorry if the question is trivial:
how could I swap even and odd elements of a 1-D 16-bit INT array
without using loops? That is, if the source array is

[a0,a1,a2,a3,..]

I want it to be

[a1,a0,a3,a2,..]

Thanks!

Subject: Re: Swap even/odd elements in array
Posted by Xavier Llobet on Wed, 22 Jun 2005 08:52:04 GMT
View Forum Message <> Reply to Message

In article <1119428174.330105.170110@o13g2000cwo.googlegroups.com>,
photosalex@freenetname.co.uk wrote:

> Hi All,
> sorry if the question is trivial:
> how could I swap even and odd elements of a 1-D 16-bit INT array
> without using loops? That is, if the source array is
>
> [a0,a1,a2,a3,..]
>
> I want it to be
>
> [a1,a0,a3,a2,..]
>
> Thanks!

```
n = (size(a))(1)
ind=2*indgen(n)
b=reform(transpose(reform([a(ind+1),a(ind)],n/2,2)),n)
```

Bear my IDL 3.6.1 notation...

--
_xavier
--
Only one "o" in my e-mail address

--
A: Because it messes up the order in which people normally read text.
Q: Why is top-posting such a bad thing?
A: Top-posting.
Q: What is the most annoying thing on usenet and in e-mail?

Subject: Re: Swap even/odd elements in array
Posted by K. Bowman on Wed, 22 Jun 2005 13:22:43 GMT
View Forum Message <> Reply to Message

In article <1119428174.330105.170110@o13g2000cwo.googlegroups.com>,
 photosalex@freenetname.co.uk wrote:

> Hi All,
> sorry if the question is trivial:
> how could I swap even and odd elements of a 1-D 16-bit INT array
> without using loops? That is, if the source array is
>
> [a0,a1,a2,a3,..]
>
> I want it to be
>
> [a1,a0,a3,a2,..]
>
> Thanks!

```
IDL> x = lindgen(16)
IDL> print, x
       0       1       2       3       4       5
6       7
       8       9      10      11      12      13
14      15
IDL> x = reform(reverse(reform(x, 2, 8), 1), 16)
IDL> print, x
       1       0       3       2       5       4
7       6
       9       8      11      10      13      12
15      14
```

Cheers, Ken Bowman

Subject: Re: Swap even/odd elements in array
Posted by David Fanning on Wed, 22 Jun 2005 14:06:46 GMT
View Forum Message <> Reply to Message

Ken Bowman writes:

>> how could I swap even and odd elements of a 1-D 16-bit INT array
>> without using loops?
>
> IDL> x = lindgen(16)
> IDL> print, x
> 0 1 2 3 4 5
> 6 7
> 8 9 10 11 12 13
> 14 15
> IDL> x = reform(reverse(reform(x, 2, 8), 1), 16)
> IDL> print, x
> 1 0 3 2 5 4
> 7 6
> 9 8 11 10 13 12
> 15 14

Those of you who have been invited to answer the Test Question
(http://www.dfanning.com/misc_tips/iepa.html) at this year's
IEPA conference might want to take careful note of this answer.
I'd say the chances are pretty good you might see it again. ;-)

Cheers,

David

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

Subject: Re: Swap even/odd elements in array
Posted by Peter Mason on Wed, 22 Jun 2005 21:43:18 GMT
View Forum Message <> Reply to Message

photosalex@freenetname.co.uk wrote:
> Hi All,
> sorry if the question is trivial:
> how could I swap even and odd elements of a 1-D 16-bit INT array
> without using loops?

One from the south-east:

```
j=indgen(16)
print,fix(swap_endian(long(swap_endian(j),0,8),0,16)
```

Subject: Re: Swap even/odd elements in array
Posted by Craig Markwardt on Thu, 23 Jun 2005 07:52:36 GMT
View Forum Message <> Reply to Message

"Peter Mason" <peter.mason@deleteme.csiro.au> writes:
> photosalex@freenetname.co.uk wrote:
>> Hi All,
>> sorry if the question is trivial:
>> how could I swap even and odd elements of a 1-D 16-bit INT array
>> without using loops?
>
> One from the south-east:
> j=indgen(16)
> print,fix(swap_endian(long(swap_endian(j),0,8),0,16)

... and a sure route to insanity. :-)

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---

Subject: Re: Swap even/odd elements in array
Posted by JD Smith on Tue, 28 Jun 2005 19:00:49 GMT
View Forum Message <> Reply to Message

On Wed, 22 Jun 2005 10:52:04 +0200, Xavier Llobet wrote:

> In article <1119428174.330105.170110@o13g2000cwo.googlegroups.com>,
> photosalex@freenetname.co.uk wrote:
>
>> Hi All,
>> sorry if the question is trivial:
>> how could I swap even and odd elements of a 1-D 16-bit INT array
>> without using loops? That is, if the source array is
>>
>> [a0,a1,a2,a3,..]
>>
>> I want it to be
>>
>> [a1,a0,a3,a2,..]
>>
>> Thanks!
>

> n = (size(a))(1)
> ind=2*indgen(n)
> b=reform(transpose(reform([a(ind+1),a(ind)],n/2,2)),n)

A similar approach using the new stride indexing operator would be:

b=reform(transpose([[a[1:*:2]],[a[0:*:2]]]),n_elements(a))

The problem with all of these dimensional-juggling approaches is that they fail for vectors of odd length (or, more generally, require the length be a strict multiple of p, where p is the transposition length).

Here's a method which doesn't have this requirement:

```
ind=lindgen(n_elements(a))/2*2 & ind[0:*:2]+=1
b=a[ind]
```

It can be generalized for any p like this:

```
ind=lindgen(n_elements(a))/p*p
for i=0,p-2 do ind[i:*:p]+=p-1-i
b=a[ind]
```

It does require the vector to have at least p-1 elements.

JD
