2-D Arrays

```c
int a[3][4];
for(i = 0; i < 3; i++)
    for(j = 0; j < 4; j++)
        a[i][j] = 0;
```

Rowwise Operations

For rowwise operations, the column subscript varies faster

```c
int a[3][4];
for(i = 0; i < 3; i++)
    for(j = 0; j < 4; j++)
        cin >> a[i][j];
```

Note that rowwise input/output is the "natural" way to do it!
Columnwise Operations

For columnwise operations, the row subscript varies faster

```c
int a[3][4];
for(j = 0; j < 4; j++)
    for(i = 0; i < 3; i++)
        cin >> a[i][j];
```

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

What would this output look like?

```c
for(j = 0; j < 4; j++)
    {
        for(i = 0; i < 3; i++)
            cout << a[i][j] << " ";
        cout << endl;
    }
```