CS 160
CS Orientation

More Lists in Python...
Odds and Ends

- Assignment #9 posted, but don’t freak out!!!
- Lab #8 this week will work toward finishing it.
How do we change the travel program to use a 2-d array?

<table>
<thead>
<tr>
<th>trip1</th>
<th>trip2</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td>hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>trip1</th>
<th>trip2</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td></td>
</tr>
<tr>
<td>hours</td>
<td></td>
</tr>
</tbody>
</table>

- This becomes harder and requires append to create because we don't know # of rows.
- How many trips?
- How many cols?
- travel_info = [[J*trips, [J*trips] row1], [J*trips] row2]
def get_speed():
    return int(input("Enter speed: "))

def get_hours():
    return int(input("Enter hours: "))

#we can pass a list and change it in a function
def get_travel_info(info):
    trips=int(input("how many trips: "))

    #can make it long way, instead of up/down, 2 rows and n columns
    info=[[0]*trips, [0]*trips]

    for i in range(trips):
        #This makes it up and down, n rows and 2 columns
        info.append([])
        info[i].append(int(input("Enter speed: ")))
        info[i].append(int(input("Enter hours: ")))

    #If we create it with [[0]*trips, [0]*trips]
    info[0][i]=int(input("Enter speed: "))
    info[1][i]=int(input("Enter hours: "))

def travel_distance(speed, hours):
Our travel program...

```python
info[i].append(int(input("Enter speed: ")))
info[i].append(int(input("Enter hours: ")))

# If we create it with [[0]*trips, [0]*trips]
info[0][i]=int(input("Enter speed: "))
info[1][i]=int(input("Enter hours: "))

def travel_distance(speed, hours):
    # for h in range(hours):
    #     print((h+1)*speed)
    for h in range(1,hours+1):
        print(h*speed)

def main():
    travel_info=[] # create a list
    get_travel_info(travel_info) # we can change a list in a function
    for i in range(len(travel_info)): # how many rows, trips
        travel_distance(travel_info[i][0],travel_info[i][1]) # pass list elements
    for i in range(len(travel_info[0])): # how many cols in a row
        travel_distance(travel_info[0][i],travel_info[1][i]) # pass list elements
    # travel_distance(get_speed(),get_hours())
main()
```
1-D heat diffusion

\[ f = \text{open}("heat.dat", "wb") \]

\[ \text{for all time} \quad \text{init condition} \]

\[ = \text{f.write(} \text{struct.pack(} \text{f"u", u_new[J])}) \]

\[ \text{for all segments not bound} \]

\[ u_{\text{new}} = \text{equation} \]

\[ \text{f.write(} \text{struct.pack(} \text{f"u", u_new[J])}) \]

\[ \text{print } u_{\text{new}} \]

\[ u_{\text{old}} = u_{\text{new}} \]

\[ \text{segments} = 4 \]

\[ u_{\text{old}} = [\text{init}] \times \text{segments} \]

\[ u_{\text{new}} = [\text{init}] \times \text{segments} \]

\[ u_{\text{old[J]}} = \text{left} \]

\[ u_{\text{old[segments-J]}} = \text{right} \]

\[ \text{# same for u.new} \]