Objectives for 9/1/2017 (Fri)

- Memory addresses
  - Finish exercise
- Memory layout (brief intro)

If time: code quality
Address syntax

```c
char s[3] = "Yo";
char* t = "Mi";
int u[3] = {5, 6, 7};
int n = 8;
```

Write the type of each of the following:

<table>
<thead>
<tr>
<th>C expression</th>
<th>in words</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>char[3]</td>
</tr>
<tr>
<td>t</td>
<td>char*</td>
</tr>
<tr>
<td>u</td>
<td>int[3]</td>
</tr>
<tr>
<td>&amp;n</td>
<td>int*</td>
</tr>
<tr>
<td>*(&amp;n)</td>
<td>int</td>
</tr>
<tr>
<td>*t</td>
<td>char</td>
</tr>
<tr>
<td>t[0]</td>
<td>char</td>
</tr>
<tr>
<td>&amp;(t[0])</td>
<td>char</td>
</tr>
<tr>
<td>s[0]</td>
<td>char</td>
</tr>
<tr>
<td>&amp;(s[0])</td>
<td>char*</td>
</tr>
<tr>
<td>*(&amp;(s[0]))</td>
<td>char</td>
</tr>
</tbody>
</table>

Code quality

1) Circle every violation of the code standards used in this class. Correct if possible. Do not circle anything that is not a violation.

```c
1 /* vim: set tabstop=4 shiftwidth=4 noexpandtab: */
2 #include <stdio.h>
3
4 /* MAIN FUNCTION (useless - obvious) */
5 int main(int argc, char *argv[]) {
6     char* theThing = "giraffe";
7     int Animal;
8     int num_zucchini_omelets_desired = 500;
9     bool is_first_letter = true;
10    Animal = 0;
11    while(theThing[Animal] != 0) {
12        if(is_first_letter) {
13            printf("%s ");
14            is_first_letter = false;
15        }
16        Animal++;  // (allowed only if this is part of the specified output.)
17        fputc(theThing[Animal], stdout);
18    }
19    fputc('\n', stdout);
20    return 0;  // EXIT_SUCCESS;
21 }
```
* ( & ( s[0] ) )

char

char

char

char c = 'A';

c[0]