QUIZ: recursive functions

```c
int mystery(int n) {
    if ( n == 0 || n == 1 || n == 2) return n;
    return (mystery(n-1) + mystery(n-2) + mystery(n-3)) ;
}
```

What is returned by `mystery(4)`?
QUIZ: recursive functions

```c
int mystery(int n) {
    if ( n == 0 || n == 1 || n == 2) return n;
    return (mystery(n-1) + mystery(n-2) + mystery(n-3)) ;
}
```

What is returned by `mystery(5)`?
Lesson 6

Program control
Lesson 6

SKIP Arrays: The Basics (p.120)
The **for** loop

```c
#include <stdio.h>
define MAXCOUNT 20
int count;

int main( void )
{
    /* Print the numbers 1 through 20 */
    for (count = 1; count <= MAXCOUNT; count++)
    {
        printf("%d\n", count);
    }
    return 0;
}
```
#include <stdio.h>
#define MAXCOUNT 20

int count;

int main( void )
{
    /* Print the numbers 1 through 20 */

    for (count = 1; count <= MAXCOUNT; count++)
        printf("%d\n", count);

    return 0;
}
for (count = 1; count <= 20; count++)
print ("\n%d", count);
for loop variations

```c
for (count = 100; count > 0; count--)
for (count = 0; count < 100; count += 5)

count = 1;
for ( ; count < 100; count++)

count = 1;
for ( printf("Hello, World!\n"); count < 100; count++)

for(count = 0; count < 100; )
   printf("%d", count++);

for ( ; ; )  // Infinite loop!
```
**for** loop variations

**Example 3**

```c
/* Lets user enter up to 10 integer values */
/* Values are stored in an array named value. If 99 is entered, the loop stops */
int value[10];
int ctr,nbr=0;
for (ctr = 0; ctr < 10 && nbr != 99; ctr++)
{
    puts("Enter a number, 99 to quit ");
    scanf("%d", &nbr);
    value[ctr] = nbr;
}
```

We shall learn about arrays in Lesson 8

puts (put string) is an output function, like printf
for loop variations

3. What is the value of \( x \) when the following statement is complete?

\[
\text{for} \ (x = 0; x < 100, x++);
\]

The body of this loop is empty!
4. What is the value of `ctr` when the following statement is complete?
   ```java
   for (ctr = 2; ctr < 10; ctr += 3);
   ```
Nested **for** loops

```c
void print_ttable(int outer, int inner)
{
    int a, b;
    for (a = 1; a <= outer; a++)
    {
        for (b = 1; b <= inner; b++)
        {
            printf("%d\t", a*b);
        }
        printf("\n");
    }
    return;
}
```
**Nested for loops**

```c
void print_ttable(int outer, int inner) {
    int a, b;
    for (a = 1; a <= outer; a++)
        for (b = 1; b <= inner; b++)
            printf("%d\t", a*b);
    printf("\n");
    return;
}
```

### Output

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>32</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>18</td>
<td>24</td>
<td>30</td>
<td>36</td>
<td>42</td>
<td>48</td>
<td>54</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>21</td>
<td>28</td>
<td>35</td>
<td>42</td>
<td>49</td>
<td>56</td>
<td>63</td>
<td>70</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>48</td>
<td>56</td>
<td>64</td>
<td>72</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>27</td>
<td>36</td>
<td>45</td>
<td>54</td>
<td>63</td>
<td>72</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
</tr>
</tbody>
</table>
Nested **for** loops

5. How many x’s does the following print?

```plaintext
for (x = 0; x < 10; x++)
    for (y = 5; y > 0; y--)
        puts("X");
```
10. **BUG BUSTER**: What is wrong with the following code fragment? *(MAXVALUES is not the problem!)*

```c
for (counter = 1; counter < MAXVALUES; counter++);
printf("\nCounter = %d", counter);
```
The **while** loop

```c
#include <stdio.h>
#define MAXCOUNT 20
int count;

int main( void )
{
    // Print the numbers 1 through 20
    count = 1;
    while (count <= MAXCOUNT)
    {
        printf("%d\n", count);
        count++;
    }
    return 0;
}
```

Note that we have the same 3 parts as in a **for** loop, but written differently.
The operation of a while statement.

while (condition) statements;

Evaluate condition

TRUE

Execute statements

FALSE

Done
QUIZ

Write a **while** loop that prints all the multiples of 3 between 50 and 100.

```c
count = 1;
while (count <= MAXCOUNT) {
    printf("%d\n", count);
    count++;
}
```
To do in notebook for next time:

• Read and take notes pp.121-130

• Answer end-of-chapter quizzes 5, 7
QUIZ

Write a **for** loop that prints all the **odd** numbers between 25 and 42 **in reverse order**.
Write a `while` loop that prints all the **odd** numbers between 25 and 42 in reverse order.
What does this double loop print?

for (i=0; i<100; i++)
    for (j=0; j<100; j++)
        if (i+j == 3)
            printf("sum = %d\n", i+j);
What does this double loop print?

```c
for (i=0; i<100; i++)
    for (j=i+1; j<100; j++)
        if (i+j == 3)
            printf("sum = %d\n", i+j);
```