Interesting recursive function from the lab:

5] Write a recursive function \textbf{multiply}(a, b), where a and b are both positive integers. You can only use the + and − operators, no ∗.

- Hint: Multiplication is repeated addition! We can recursively write \(a \times b = (a-1) \times b + b\). Test with \(a = 4, b = 5\).

```c
#include <stdio.h>

int mult(int a, int b){
    if (a==0)
        return 0; /* 0 times b equals 0 */
    else
        return mult(a-1, b) + b;
}

void main(void){
    printf("%d\n\n", mult(4, 5));
}
```

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;
}
```
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;
}
```

No semicolon here, because the **for** command has not yet ended!
for (count = 1; count <= 20; count++)
  print ( "\n%d", count);
**for loop variations**

```plaintext
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 ( ; ; )  
```
for loop variations

Example 3

/* 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;
}

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

We shall learn about arrays in Lesson 8
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!
for loop variations

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;
}
```
### Output ▼

The times table:

<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>

```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

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);
```
Extra-credit
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
FIGURE 6.3
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++;
}
```
What does this double loop print?

for (i=10; i<100; i++)
    for (j=10; j<100; j++)
        if (i+j == 13)
            printf("sum = %d\n", i+j);
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.
QUIZ

Write a `while` loop that prints all the odd numbers between 25 and 42 in reverse order.

```c
count = 1;
while (count <= MAXCOUNT) {
    printf("%d\n", count);
    count++;
}
```
What does this double loop print?

```c
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);
```
The **do ... while** loop

```c
int selection = 0;

do {
    printf("\n ");
    printf("\n1 - Add a Record ");
    printf("\n2 - Change a record ");
    printf("\n3 - Delete a record ");
    printf("\n4 - Quit ");
    printf("\n ");
    printf("\nEnter a selection: ");

    scanf("%d", &selection );

} while ( selection < 1 || selection > 4 );
```

Note that we have the same 3 parts as in a **for** loop, but written differently

**Output ▼**

```
1 - Add a Record
2 - Change a record
3 - Delete a record
4 - Quit

Enter a selection: 8

Enter a selection: 4
You chose Menu Option 4
```
do statements;
while (condition);

Evaluate condition

TRUE

FALSE

Done
QUIZ

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

Write a do ... while loop that prints all the odd numbers between 25 and 42 in reverse order.
Important when working with loops

Check the boundary conditions (where the loop starts and where it ends) carefully.

Make sure you understand:

• What happens the first time through the loop
• What happens the last time through the loop
• How many times the body of the loop has executed
• What are the values of the variables after the loop has exited
What is the difference between these 2 types of loops?
To which of these 2 types does each of the 3 C loops belong?
How to choose the appropriate loop for the task at hand

First decision: test at the end or at the beginning?

• at end → do ... while

• at beginning:
  • Know # of iterations → for
  • Don’t know → while
Nested loops

The right way
Inner loop is completely inside the outer

The wrong way
Loops straddle each other
Homework for Chapter 6:

• Ch. 6 – Exercises 6, 7, 8, 9

Due Monday, Feb. 15 (but recommended as preparation for midterm!)

For each coding exercise capture a screenshot of both code and output

Give instructor one printout with the entire homework.