Section 9: Loops

Introduction: A loop allows us to execute the same block of code multiple times until a specified conditional expression becomes false (i.e. “do <something> until <condition> fails”). Similar to multiple function calls, loops tend to be most useful when they are used to execute similar (not identical) sets of instructions. You may find it helpful to think of a loop as a condensed form of repeated, similar code.

while-loops: This type of loop repeatedly runs the code inside of it while a conditional expression is true:

```
while ( condition ) {
    body;  // while-loop body
}
```

Notice how the code inside the loop is contained within curly braces, just like the code in a function! In general, curly braces denote a “block” of code.

Example:
```
int x = 1;
while ( x < 10 ) {
    x = x * 2;
}
```

The above loop will execute the statement \( x = x \times 2 \) four times, with the final value of \( x = 16 \):

<table>
<thead>
<tr>
<th>Iteration</th>
<th>x</th>
<th>Condition ((x &lt; 10))</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>true</td>
<td>Execute ( x = 1 \times 2 );</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>true</td>
<td>Execute ( x = 2 \times 2 );</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>true</td>
<td>Execute ( x = 4 \times 2 );</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>true</td>
<td>Execute ( x = 8 \times 2 );</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>false</td>
<td>Exit loop</td>
</tr>
</tbody>
</table>
Exercises:

1) Describe what the loop below does.

```c
int pos = 0;
while ( pos < min(width, height) ) {
    rect(pos, pos, 50, 50);
    pos = pos + 50;
}
```

2) Complete the loop below to find the **smallest power of 3 greater than 100**. Your answer should be stored in the variable `answer` after the loop has executed:

```c
int answer = _____;
while ( __________________________ ) {
    answer = ______________________;
}
```

3) Complete the loop below that calculates the **sum of all even integers from 50 to 100, inclusive**. Your answer should be stored in the variable `sum` after the loop has executed:

```c
int sum = _____;
int i = _____;
while ( i <= _____ ) {
    sum = ______________________;
    i = i + _____;
}
```

4) Find a partner, brainstorm Creativity Project ideas, and get started on “Creativity Planning.”