Lab 3

Saad Mneimneh
Hunter College of CUNY

Practice loops and nested loops

**Lab A: Lucas numbers**

Lucas numbers are defined as the following sequence:

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>L(n)</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>18</td>
<td>...</td>
</tr>
</tbody>
</table>

Write a function called lucas that accepts \( n \geq 0 \) as an integer parameter and returns \( L(n) \).

**Lab B: Pythagoras**

A pythagorian triple is a triple \((a, b, c)\) such that \(a^2 + b^2 = c^2\) and \(a, b, c\) are integers. We will assume that \(a < b < c\). For example \((3, 4, 5)\) is a pythagorian triple. In addition, a pythagorian triple is primitive if \(a, b, c\) do not have a common divisor. For instance \((3, 4, 5)\) is primitive, but \((6, 8, 10)\) is not.

Your task is to output all primitive pythagorian triples with \(1 \leq a, b, c \leq 100\),

(a) using three nested loops  
(b) using two nested loops. *Hint: you might want to use the sqrt function by including cmath (i.e. use #include <cmath>).*

Recall the function that finds the greatest common divisor of two integers:

```cpp
int gcd(int a, int b) {
    while (b!=0) {
        int c=a;
        a=b;
        b=c%a;
    }
    return a;
}
```

**Lab C: Perfect numbers**

A number is perfect if it is equal to the sum of all its proper divisors. For example, 6 is perfect because the divisors of 6 are \{1,2,3,6\} and \(6=1+2+3\). Find the first 4 perfect numbers (including 6).