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 \( L(n) \).

```c++
int lucas(int n) {
    int a=-1;
    int b=2;
    for (int i=1; i<=n; i=i+1) {
        int c=a;
        a=b;
        b=c+b;
    }
    return b;
}
```

Lab B: Pythagoras
A pythagorean triple is a triple \((a, b, c)\) such that \(a^2 + b^2 = c^2\) and \(a, b,\) and \(c\) are integers. We will assume that \(a < b < c\). For example \((3, 4, 5)\) is a pythagorean triple. In addition, a pythagorean triple is primitive if \(a, b,\) and \(c\) 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 pythagorean triples with \(1 \leq a, b, c \leq 100\),

(a) using three nested loops

```c++
int main() {
    for (int a=1; a<=100; a=a+1)
        for (int b=a+1; b<=100; b=b+1)
            for (int c=b+1; c<=100; c=c+1)
                if (c*c==a*a+b*b && gcd(a,gcd(b,c))==1)
                    cout<< '('<<a<<','<<b<<','<<c<<')'<<'
';
}
```
(b) using two nested loops. *Hint:* you might want to use the sqrt function by including cmath (i.e. use `#include <cmath>`).

```cpp
int main() {
    for (int a=1; a<=100; a=a+1)
        for (int b=a+1; b<=100; b=b+1) {
            int c=sqrt(a*a+b*b);
            if (c*c==a*a+b*b && gcd(a,gcd(b,c))==1)
                cout<<("('<<a<<','<<b<<','<<c<<")\n";
        }
}
```

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).

```cpp
int main() {
    int found=0;
    for (int n=1; found<4; n=n+1) {
        int s=0;
        for (int d=1; d<n; d=d+1)
            if (n%d==0)
                s=s+d;
        if (s==n) { //n is perfect
            cout<<("n\n";
            found=found+1;
        }
    }
    // it will output:
    // 6
    // 28
    // 496
    // 8128
```