Eliminating Recursion with Stacks

CSCI 112, Lab 6

File names: Names of files, functions, and variables, when specified, must be EXACTLY as specified. This includes simple mistakes such as capitalization.

Individual work: All work must be your own. Do not share code with anyone other than the instructor and teaching assistants. This includes looking over shoulders at screens with the code open. You may discuss ideas, algorithms, approaches, *etc.* with other students but NEVER actual code. Do not use code written by anyone else, in the class or from the internet.

Documentation: Each file should begin with a docstring that includes your name, the class number and name, the lab number, and a short description of the lab, as well as documentation pertinent to that particular file.

Translations: Translate each of the following recursive functions into "unnested" versions and into non-recursive functions following the style of the functions discussed in class and available in the file `stackrecursion.py`. Put all the functions into a file called `uncursion.py` and unit tests for all of them into a file called `uncursion_test.py`.

Put everything into a folder names `csci112lab06yourname`, zip and turn into canvas.

The problems:

1.
```python
def func01(n):
    if n < 2:
        return 3*n
    else:
        return 2*func01(n-1)
```

2.
```python
def func02(n):
    if n < 2:
        return 3*n
    elif n < 6:
        return n*n
    else:
        return 3 + 2*func02(n-2)
```

3.
```python
def func03(n):
    if n < 2:
        return 3*n
    else:
        return 3*func03(n-1) + 2*func03(n-2)
```

4.
```python
def func04(n):
    if n < 2:
        return 3*n
    elif n%2 == 1:
        return 7 + func04(n - 3)
    else:
        return func04(n-1) * 2*func04(n//2)
```

5.
```
def func05(a, b):
    if a == 0:
        return b+3
    elif b == 0:
        return a*2
    else:
        return 2*a + 3*b + 4*func05(a-1,b) + 5*func05(a, b-1)
```