Concurrent Programming[1]

Exercise Booklet 7: Erlang – Sequential Fragment

1. Write the following functions in Erlang

   a) mult

   b) double

   c) distance

   d) and

   e) or

   f) not

2. What is the result of typing these two lines?

```
1> {A,B} = {2,3}.
2> B.
```

3. What is the result of these two lines, if they're typed after the previous two?

```
3> {A,C} = {2,5}.
4> {A,D} = {6,6}.
```

4. What is the output of each of these lines?

```
1> A=2+3.
2> B=A-1.
3> A=B+1.
4> A=B.
```

5. What is the output of each of these lines?

```
5> f(A).
6> A=B.
7> f().
```

6. Implement the following functions:

   a) fibonacci

   b) fibonacciTR: tail recursive fibonacci

7. Implement the following functions

   a) sum

   b) maximum

   c) zip

   d) append

   e) reverse

   f) evenL: returns the sublist of even numbers in a given list of numbers

   g) take

[1] Some exercises are taken from Simon Thompson's online tutorial on Erlang.

h) drop

8. Type this out in a file `test.erl`.

```
-export([take/2]).
-include_lib("eunit/include/eunit.hrl").

take(_,[]) -> [].

take_1_test() ->
   ?assertEqual(take(0,[]),[]).
take_2_test() ->
    ?assertEqual(take(0,[1,2,3]),[]).
```

Then type out the following in a shell and write down the output:

```
1> c(test).
{ok,test}
2> test:test().
```

9. Define in Erlang

   a) map

   b) filter

   c) fold

10. Represent binary trees using tuples:

    - `{empty}` and
    - `{node,aNumber,lsubtree,rsubtree}`.

    Then implement:

    a) mapTree

    b) foldTree