# E-402-STFO PROBLEMS FOR MODULE 3

CREATED BY HENNING ULFARSSON

This is the first module concerned with dynamic programming. You get a perfect score for this module by getting 55 points or more.

## 1. Rabbits (26 points)

The *Tribonacci numbers* $t(n)$ are defined for non-negative integers as follows:

$$t(0) = t(1) = 0,\ t(2) = 1,\ t(n) = t(n-1) + t(n-2) + t(n-3) \text{ for } n \geq 3.$$

In the next four problems we will explore different ways to compute the values of these numbers. The functions take as input a positive integer $n$ and return the $n$-th Tribonacci number.

**Problem 1** (2 points)**.** Write a recursive function `m3p1` by filling in the missing code in the function below.

```
def m3p1(n):

    if n <= 1:
        pass
    if n == 2:
        pass

    pass
```

For example:

```
Input: m3p1(8)
Output: 24
```

This version should start getting too slow at around $n = 30$.

**Problem 2** (4 points)**.** Write a non-recursive function `m3p2` by filling in the missing code in the function below. It should use a dictionary to store values already computed.

```
memo = {0:0, 1:0, 2: 1}

def m3p2(n):

    if n in memo:
        pass

    else:
        pass
```

This version should break at exactly $n = 996$ because that will exceed the default maximum recursion depth of Sage (at least on my machine).

*Date*: Updated September 15, 2013.

**Problem 3** (7 points)**.** Write a non-recursive function `m3p3` by filling in the missing code in the function below. It should use a list to store values already computed.

```
def m3p3(n):

    # Initializing a list to contain the values
    T = [0,0,1] + [0]*(n-2)

    for i in range(3,n+1):
        pass

    return T[n]
```

This version should handle values such as $n = 10^5$, but will start getting too slow soon after that.

**Problem 4** (13 points)**.** Write a non-recursive function `m3p4` by filling in the missing code in the function below. It should only store three values.

```
def m3p4(n):

    if n <=1 :
        return 0

    t3, t2, t1 = 1,0,0

    for i in range(3,n+1):
        pass

    return SOMETHING :)
```

This version should handle values such as $n = 5 * 10^5$, but will start getting too slow soon after that.

## 2. Project Euler Problems (55 points)

Please note: You are not allowed to use the built-in function `Partitions(n)` (or any other built-in function with similar capabilites) to solve the problems in this section.

**Problem 5** (10 points)**.** Write a function `m3p5(n,L)` that solves Project Euler problem 31 (`http://projecteuler.net/problem=31`). The input `n` is a positive integer and the input `L` is a list of coin values. The returned value is the number of ways `n` can be split using the coin values in the list `L`.

```
Input: m3p5(5,[1,2,3])
Output: 5
```

**Problem 6** (10 points)**.** Write a function `m3p6(k)` that solves Project Euler problem 76 (`http://projecteuler.net/problem=76`). The input `k` should be a positive integer. The returned value is the number of different ways `k` can be written as a sum of at least two positive integers.

```
Input: m3p6(5)
Output: 6
```

**Problem 7** (10 points)**.** Write a function `m3p7(k)` that solves Project Euler problem 77 (`http://projecteuler.net/problem=77`). The input `k` should be a positive integer. The returned value is the smallest positive integer $n$ such that the number of ways to write $n$ as a sum of primes exceeds `k`.

```
Input: m3p7(4)
Output: 10
```

**Problem 8** (10 points)**.** Write a function `m3p8(k)` that solves Project Euler problem 78 (`http://projecteuler.net/problem=78`). The input `k` should be a positive integer. The returned value is the smallest positive integer $n$ such that number of ways $n$ coins can be separated into piles is divisible by `k`

```
Input: m3p8(7)
Output: 5
```

**Problem 9** (15 points)**.** Write a function `m3p9(M)` that solves Project Euler problem 81 (`http://projecteuler.net/problem=81`). The input `M` should be an $n \times n$ matrix containing integers.

```
Input: m3p9(matrix[[9,9,1], [1,9,9], [1,1,9]])
Output: 45
```

School of Computer Science, Reykjavik University, Menntavegi 1, 101 Reykjavík, Iceland

*E-mail address*: `henningu@ru.is`