General Computer Science II (320102) Spring 2014
Assignment 5: ASM Language
(Given Mar. 5., Due Mar. 12.)

Problem 5.1 (Convert Highlevel Code to $\mathcal{L}$(VM) Code)

Given is an array `A[0..10]` and the following piece of imperative code: 20pt

```
for j := 1 to 5 do
 for i := j to 10-j do
  A[i] := A[i-j] + A[i+j];
```

Suppose the array is loaded on stack (top value being `A[10]`). Convert the code into $\mathcal{L}$(VM) code.

Problem 5.2 (Simulating REMA in SML)

40pt

Given the following declarations:

```
datatype register = acc | in1 | in2;
datatype instr = load of int | loadi of int | loadin1 of int | loadin2 of int |
                 store of int | storein1 of int | storein2 of int |
                 add of int | addi of int | sub of int | subi of int |
                 move of register*register| nop of int | stop of int |
                 jump of int | jumpe of int | jumpne of int |
                 jumpl of int | jumple of int | jumpg of int | jumpge of int;
type program = instr list;
type memory = int list;

(* This is the state of the machine. From left to right the values mean:
   PC register; ACC register; IN1 register; IN2 resigter; Memory cells*)
type state = int*int*int*int*(int list);
```

Write two SML functions:

- `execute_instr : instr -> state -> state`
- `run : program -> memory -> memory`

The first function takes an `ASM` instruction and the current state of the `REMA` as arguments and returns the new state after the instruction is executed. The second function takes a program and the initial configuration of the memory. It then simulates the program until a `STOP 0` instruction is reached and returns the memory at that point. In both functions 'memory' is just a list of integers that represent the current state of the memory of the `REMA`. Once the initial list is supplied, during simulation its length shoudn't change.

Note: For this problem and the next it will be very helpful to use built-in SML functions. Make sure to check the forums for more info.

Problem 5.3 (SW TCN Converter)

Write a Simple While program that converts TCN to decimal. You will be given the TCN number t (consider it a variable set at the beginning of your program). You can consider 25pt

that the number t is an integer formed only from 1s and 0s and the most significant bit is represented by the first digit; remember that the first bit also tells you whether the number is negative or positive.

Hint: Use the formula for conversion between TCN and decimal!

Your program should start, for example, with:

```
var t=11001;
```

That is, $t = 11001$, which corresponds to -7.

Problem 5.4 (Simple While program on Fibonacci) 15pt

Write a Simple While Program that takes a number N and computes the N^{th} Fibonacci number. Then provide the Abstract Syntax for your code.

Show how the $\mathcal{L}$(VM) version of it looks like by compiling it.

Hint: Remember that the n^{th} Fibonacci number is given by the following recursive equations:

$$fib(n+1) = f(n) + fib(n-1) \qquad fib(0) = 0 \qquad fib(1) = 1$$