

## CS 3721: Programming Languages Lab

### Lab #07: Assignments and loops in ML

ML supports variable assignments through the "==" operator, and only reference values (i.e., storages associated with pointer values) can be modified. For example, if we want to associate variable *myvar* with a memory storage that can be modified using assignment, we can define *myvar* as the following.

```
- val myvar = ref 2.0;
val myvar = ref 2.0 : real ref
- myvar := 3.5;
val it = () : unit
```

The following example uses a sequence of statements to modify reference storages before returning a result.

```
let val a = ref 2.0; val b = ref 3.0
in
  a := !a + !b;
  b := !a * !a;
  b
end;
```

ML also supports loops in the following syntax.

```
while <expression> do <expression>
```

The following illustrates the use of while-do loops.

```
val i = ref 1;
while !i <= 10 do i := !i + 1;
```

Note that *!i* gets the value associated with variable *i*.

1. Define an integer reference variable *myvar* which initially has value 1; increment the value stored in *myvar* by 5; then return the value of *myvar* as result.
  
2. Define a reference variable *mylist* which initially has empty list as its value. Modify *mylist* by prepending strings "a" and "b" to the list. Then, return the value of *mylist* as result.

3. Define a reference variable *mylist* which initially has empty list as its value. Write a loop which modifies *mylist* by prepending integers 1 through 50 to the list. Finally, return the value of *mylist* as result.

4. Write a ML function *OddSum* which takes a single integer parameter *x* and returns the sum of all the odd integer numbers between 1 and *x*. You could use the following test cases for your code.

```
- OddSum(10);  
val it = 25 : int  
- OddSum(15);  
val it = 64 : int  
- OddSum(30);  
val it = 225 : int
```

5. The following defines the factorial function recursively.

```
fun fact(n) = if n = 1 then 1 else n * fact(n-1);
```

Define a new function *fact1* which computes the factorial of a number using the while loop.