Daniel Frey
CS 3160-001
Assignment 3
10/1/18

1. Correct expression in ML? Give type if valid.
   a. null []; **VALID : bool**
   b. hd [7, "c"]; **INVALID**
   c. tl(3::[4,5]); **VALID : int list**
   d. #First {First=3, next="mine"} **with ';' VALID : int**
   e. fn x=> (ref x) = 5 ; **INVALID**
   f. 5 + 7.9 **INVALID**
   g. nil **with ';' VALID : 'a**
   h. !(ref 7) + 8 **with ';' VALID : int**

2. ML function **Append** to append two lists.

```
fun Append ([], L2) = L2
  | Append (L1, []) = L1
  | Append ((hL1 :: tL1), (hL2 :: tL2)) =
      if hL1 < hL2
      then hL1 :: (Append (tL1, (hL2 :: tL2)))
      else hL2 :: (Append ((hL1 :: tL1), tL2));
```

Append([1,3,5],[2,4,6]);
val it = [1,2,3,4,5,6] : int list

```
- fun Append ([], L2) = L2
=   | Append (L1, []) = L1
=   | Append ((hL1::tL1), (hL2::tL2)) =
= if hL1 < hL2
= then hL1 :: (Append (tL1, (hL2::tL2)))
= else hL2 :: (Append ((hL1::tL1), tL2));
val Append = fn : int list * int list -> int list
- Append([1,3,5],[2,4,6]);
val it = [1,2,3,4,5,6] : int list
```

3. Translate Lisp to ML.

```
(defun mult (x)(cond ((null x) 1)
        ((consp x) (* (car x) (mult (cdr x))))))
(print (mult '(4 5 6)))
```

ML:

```
fun mult ([]) = 1
  | mult (hL :: tL) = hL * mult(tL);
```

4. ML program that outputs list of first elements of every list of tuples.

```
fun first ([]) = []
   | first ((e1, e2) :: tL) = e1 :: first(tL);
```

first([1, “How”), (2, “Are”), (3, “you”)]);
val it = [1,2,3] : int list

```
- fun first ([]) = []
=   | first ((e1, e2) :: tL) = e1 :: first(tL);
val first = fn : ('a * 'b) list -> 'a list
- first([(1, "How"), (2, "Are"), (3, "you")]);
val it = [1,2,3] : int list
```

5. ML program to produce n-th Fibonacci number.

```
fun Fib (n) =
        if n < 3 then 1
        else Fib(n-1) + Fib(n-2);
```

```
- fun Fib (n) =
= if n < 3 then 1
= else Fib(n-1) + Fib(n-2);
val Fib = fn : int -> int
- Fib 6;
val it = 8 : int
- Fib 7;
val it = 13 : int
```