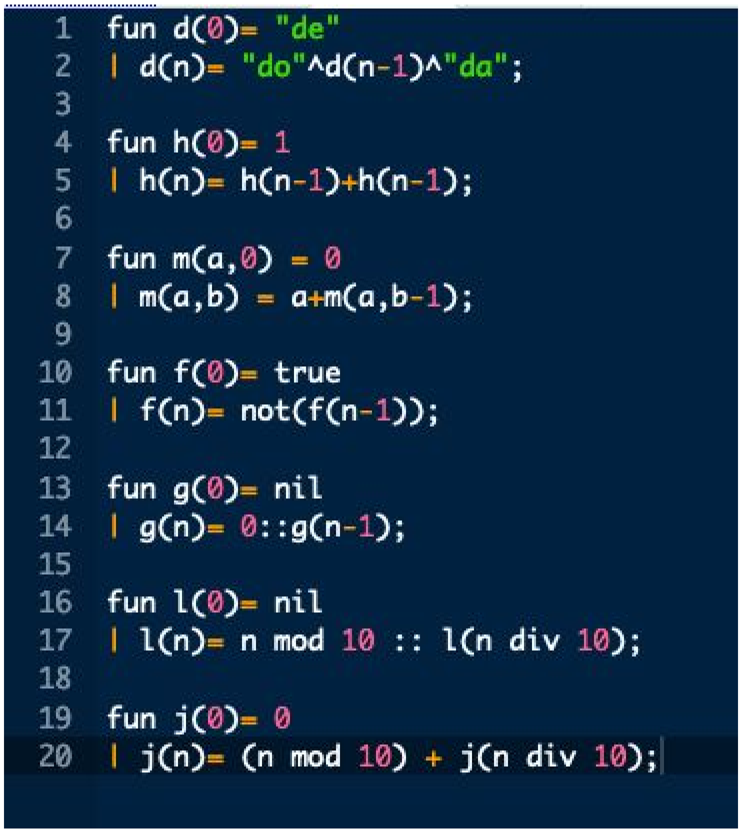
Csc 301 Lab 5
Baheem Ferrell

GIML: Tutorial Three

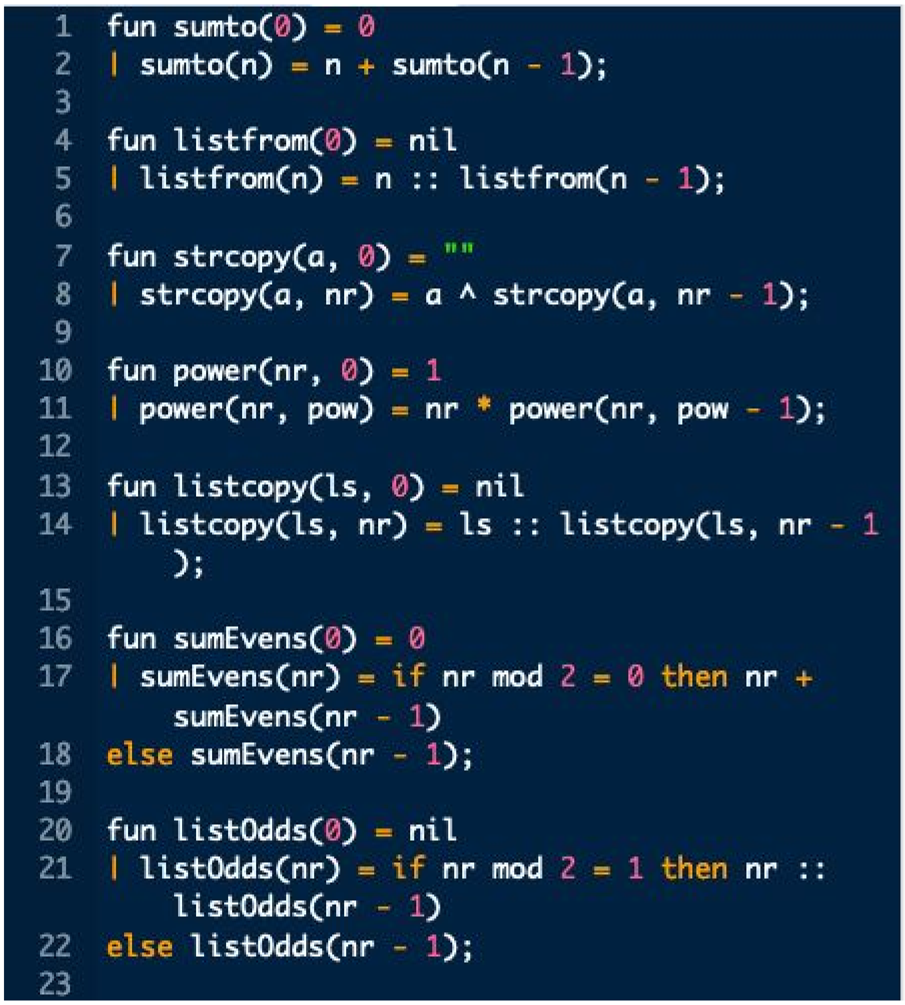
1. Paste each of the following function definitions into ML and evaluate each a a few values. Be sure that you can see



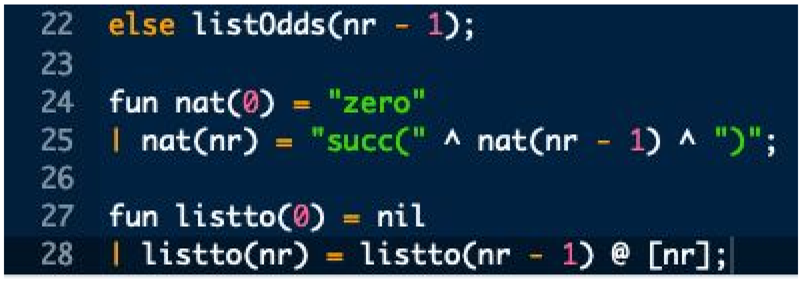
```
fun d(0)= "de"
| d(n)= "do"^d(n-1)^"da";

fun h(0)= 1
| h(n)= h(n-1)+h(n-1);

fun m(a,0) = 0
| m(a,b) = a+m(a,b-1);

fun f(0)= true
| f(n)= not(f(n-1));

fun g(0)= nil
| g(n)= 0::g(n-1);

fun l(0)= nil
| l(n)= n mod 10 :: l(n div 10);

fun j(0)= 0
| j(n)= (n mod 10) + j(n div 10);
```

```
- val d = fn : int -> string
val h = fn : int -> int
val m = fn : int * int -> int
val f = fn : int -> bool
val g = fn : int -> int list
val l = fn : int -> int list
val j = fn : int -> int
-
```

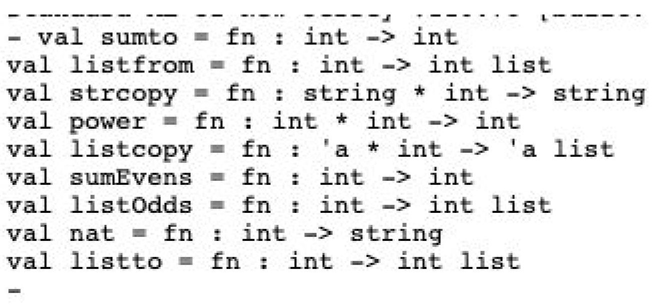
2. Each of the following functions can be defined in a similar way. An example has been given in each case:



```
fun sumto(0) = 0
| sumto(n) = n + sumto(n - 1);

fun listfrom(0) = nil
| listfrom(n) = n :: listfrom(n - 1);

fun strcopy(a, 0) = ""
| strcopy(a, nr) = a ^ strcopy(a, nr - 1);

fun power(nr, 0) = 1
| power(nr, pow) = nr * power(nr, pow - 1);

fun listcopy(ls, 0) = nil
| listcopy(ls, nr) = ls :: listcopy(ls, nr - 1);

fun sumEvens(0) = 0
| sumEvens(nr) = if nr mod 2 = 0 then nr + sumEvens(nr - 1)
else sumEvens(nr - 1);

fun listOdds(0) = nil
| listOdds(nr) = if nr mod 2 = 1 then nr :: listOdds(nr - 1)
else listOdds(nr - 1);

```



```
else listOdds(nr - 1);

fun nat(0) = "zero"
| nat(nr) = "succ(" ^ nat(nr - 1) ^ ")";

fun listto(0) = nil
| listto(nr) = listto(nr - 1) @ [nr];
```



```
- val sumto = fn : int -> int
val listfrom = fn : int -> int list
val strcopy = fn : string * int -> string
val power = fn : int * int -> int
val listcopy = fn : 'a * int -> 'a list
val sumEvens = fn : int -> int
val listOdds = fn : int -> int list
val nat = fn : int -> string
val listto = fn : int -> int list
-
```