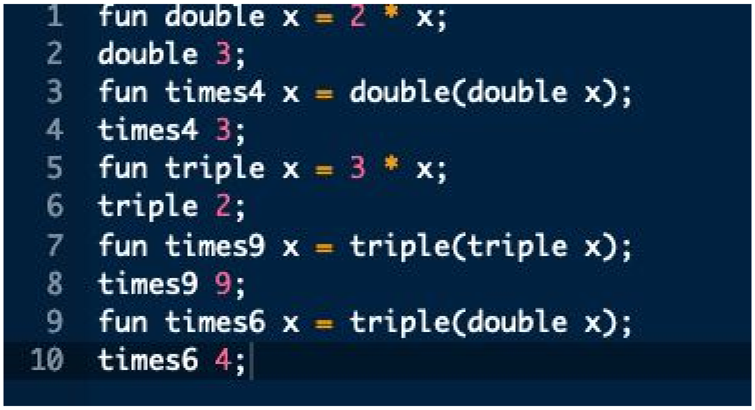
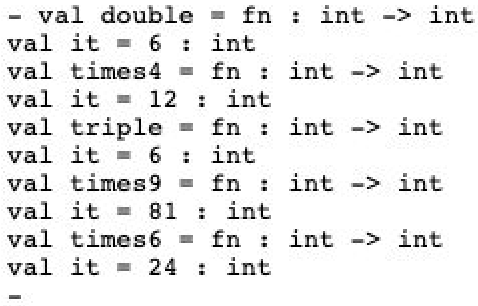
301Lab 3: Basic ML

Baheem Ferrell

1. Define and test the functions double and triple. double is given below: fun double x = 2 * x; This function may be exercised by entering an expression to be evaluated. For example double 3; The function times4 may be defined by applying double twice. This is function composition.

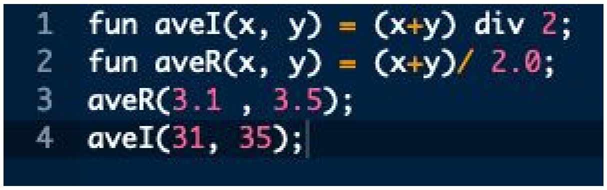


```
fun double x = 2 * x;
double 3;
fun times4 x = double(double x);
times4 3;
fun triple x = 3 * x;
triple 2;
fun times9 x = triple(triple x);
times9 9;
fun times6 x = triple(double x);
times6 4;
```

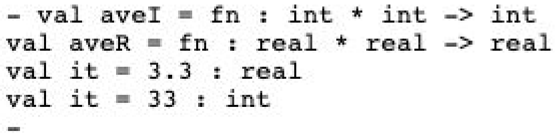


```
- val double = fn : int -> int
val it = 6 : int
val times4 = fn : int -> int
val it = 12 : int
val triple = fn : int -> int
val it = 6 : int
val times9 = fn : int -> int
val it = 81 : int
val times6 = fn : int -> int
val it = 24 : int
-
```

2. Functions with more than one input may be defined using "tuples". Define and test aveI and aveR given: fun aveI(x,y) = (x+y) div 2; fun aveR(x,y) = (x+y)/2.0; Notice how ML works out the type of each function for itself. Try… aveR(3.1 , 3.5); aveI(31, 35);

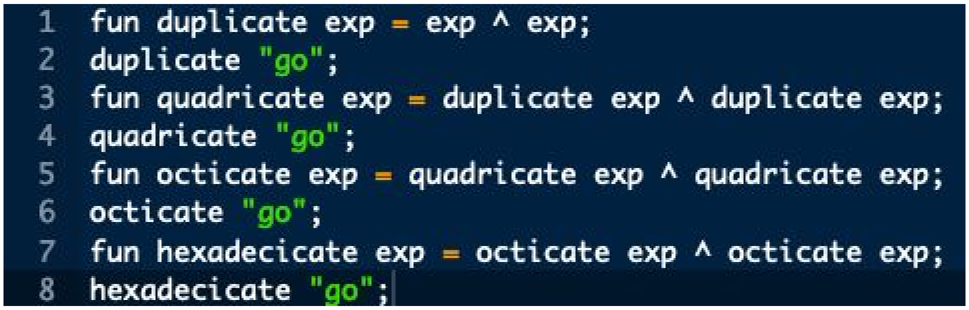


```
fun aveI(x, y) = (x+y) div 2;
fun aveR(x, y) = (x+y)/ 2.0;
aveR(3.1 , 3.5);
aveI(31, 35);
```

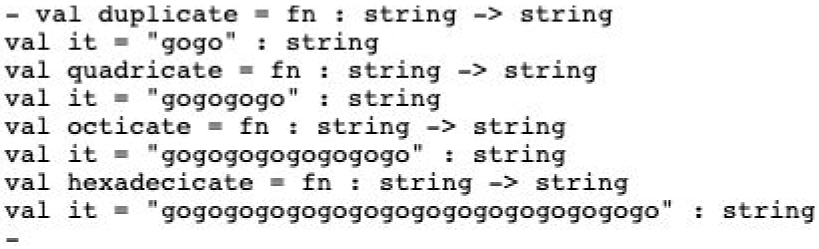


```
- val aveI = fn : int * int -> int
val aveR = fn : real * real -> real
val it = 3.3 : real
val it = 33 : int
-
```

3. Evaluate the expression "one"^"one" . Define the function duplicate such that duplicate "go" evaluates to "gogo" Also define quadricuadricate, octicate and hexadecicate.

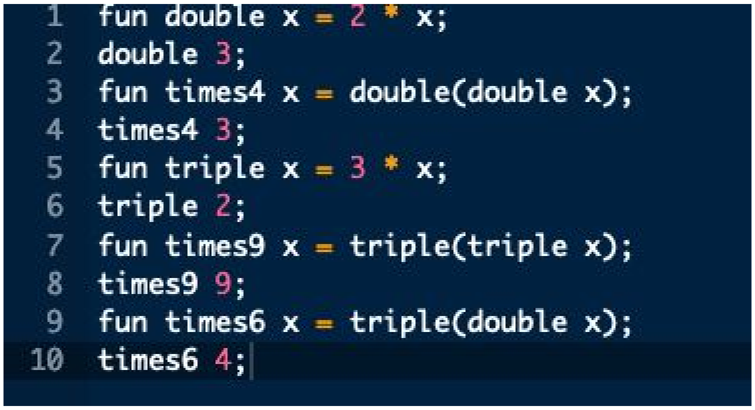


```
fun duplicate exp = exp ^ exp;
duplicate "go";
fun quadricate exp = duplicate exp ^ duplicate exp;
quadricate "go";
fun octicate exp = quadricate exp ^ quadricate exp;
octicate "go";
fun hexadecicate exp = octicate exp ^ octicate exp;
hexadecicate "go";
```

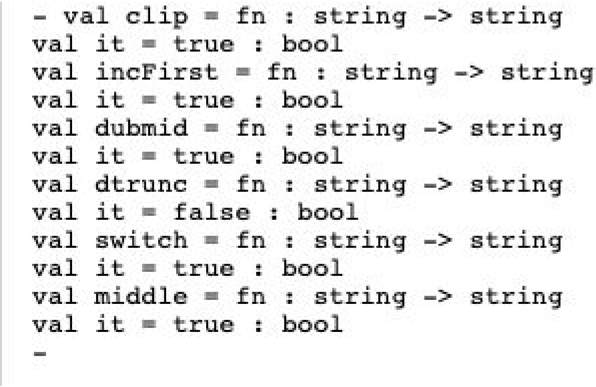


```
- val duplicate = fn : string -> string
val it = "gogo" : string
val quadricate = fn : string -> string
val it = "gogogogo" : string
val octicate = fn : string -> string
val it = "gogogogogogogogo" : string
val hexadecicate = fn : string -> string
val it = "gogogogogogogogogogogogogogogogo" : string
-
```

4. Define the following functions given by example here:



```
fun double x = 2 * x;
double 3;
fun times4 x = double(double x);
times4 3;
fun triple x = 3 * x;
triple 2;
fun times9 x = triple(triple x);
times9 9;
fun times6 x = triple(double x);
times6 4;
```



```
- val clip = fn : string -> string
val it = true : bool
val incFirst = fn : string -> string
val it = true : bool
val dubmid = fn : string -> string
val it = true : bool
val dtrunc = fn : string -> string
val it = false : bool
val switch = fn : string -> string
val it = true : bool
val middle = fn : string -> string
val it = true : bool
-
```