

## Quiz 2

### Questions 1 - 3 (6 versions for #1-2, 4 versions for #3)

```
1. fun pair_up f g x = (f x, g x)
   val n = ???
   val z = pair_up (fn x => f') (fn x => g') n
   val ans = (z = z')
```

<b>f'</b>	<b>g'</b>	<b>z'</b>	<b>n</b>
$x * 2$	$x + 1$	(4, 3)	2
$x * 2$	$x + 1$	(16, 9)	8
$x * 2$	$x + 1$	(10, 6)	5
$x + 2$	$x * 3$	(11, 27)	9
$x + 2$	$x * 3$	(3, 3)	1
$x + 2$	$x * 3$	(9, 21)	7

```
2. val xs = xs'
   val f = ???
   val ys = List.map f xs
   val ans = (ys = ys')
```

<b>xs'</b>	<b>ys'</b>	<b>f</b>
[8, 6, 7, 5, 3, 0, 9]	[16, 12, 14, 10, 6, 0, 18]	$\text{fn } x \Rightarrow x * 2$
[8, 6, 7, 5, 3, 0, 9]	[11, 9, 10, 8, 6, 3, 12]	$\text{fn } x \Rightarrow x + 3$
[8, 6, 7, 5, 3, 0, 9]	[7, 5, 6, 4, 2, ~1, 8]	$\text{fn } x \Rightarrow x - 1$
[4, 8, 15, 16, 23, 42]	[8, 16, 30, 32, 46, 84]	$\text{fn } x \Rightarrow x * 2$
[4, 8, 15, 16, 23, 42]	[7, 11, 18, 19, 26, 45]	$\text{fn } x \Rightarrow x + 3$
[4, 8, 15, 16, 23, 42]	[3, 7, 14, 15, 22, 41]	$\text{fn } x \Rightarrow x - 1$

```
3. val f' = List.foldl z' 0
   val y = f' [1, 2, 3, 4, 5, 6]
   val ans = (y = y')
```

f'	y'	z'
sum_evens	12	fn (x, y) => if (x mod 2 = 0) then x + y else y
count_evens	3	fn (x, y) => if (x mod 2 = 0) then 1 + y else y
sum_odds	9	fn (x, y) => if (x mod 2 = 1) then x + y else y
count_odds	3	fn (x, y) => if (x mod 2 = 1) then 1 + y else y

#### Question 4 (3 versions)

```
4. (* duplicate n s produces a string consisting of the string s repeated n
   * times, assuming n >= 0 *)
   fun duplicate n s = n_times ??? n "" (* a *)

   (* triple_string s produces a string consisting of the string s repeated 3
   * times *)
   val triple_string = ??? (* b *)

   a. (fn x => x ^ s)
   b. duplicate 3
```

---

```
(* mult x y computes x times y, assuming y >= 0 *)
fun mult x y = n_times ??? y 0 (* a *)

(* ten_times n computes 10 times n, assuming n >= 0 *)
val ten_times = ??? (* b *)

   a. (fn n => n + x)
   b. mult 10
```

---

```
(* pow x y computes x to the yth power, assuming y >= 0 *)
fun pow x y = n_times ??? y 1 (* a *)

(* two_to_the n computes 2 to the nth power, assuming n >= 0 *)
val two_to_the = ??? (* b *)

   a. (fn n => n * x)
   b. pow 2
```

**Questions 5-8 (each student saw 4 of 5)**

<pre> fun b1 f g x = f (g x)  fun b2 f g = fn x =&gt; (f o g) x </pre>	ALWAYS
<pre> fun b1 f x = f x + f 2  fun b2 f x =   let val z = f 2   in f x + z   end </pre>	PURE
<pre> fun b1 f y = fn x =&gt; f x + y  fun b2 f y = fn q =&gt; f q + y </pre>	ALWAYS
<pre> fun b1 f g = fn x =&gt; f x andalso g x  fun b2 f g =   fn x =&gt;     Let       val v1 = f x       val v2 = g x     In       v1 andalso v2     end </pre>	PURE
<pre> fun b1 f g xs = List.map f (List.filter g xs)  fun b2 f g xs = List.filter g (List.map f xs) </pre>	NEVER

**Question 9**

```

datatype nato = Alpha of bool
              | Bravo of int
              | Charlie of string * nato
              | Delta of nato list

```

**Question 10**

```

signature CLOCKTIME =
sig
  type time
  exception BadTime

  val midnight : time
  val noon : time

  val to_string : time -> string
  val add_minutes : time * int -> time
end

```