1 ML - 14 points

What do the following ML expressions evaluate to:

```
let fun f x = x+x
in f 7
end
```

```
let fun f x = x+x
   val x = 3
in f 7
end
```

```
let val x = 3
   fun f x = x+x
in f 7
end
```

```
let val x = 3
   fun f y = x+y
in f 7
end
```
let val x = 3
  fun f y = x+y
in let val x = 5
  in f 7
  end
end

let val g = let val x = 3
  fun f y = x+y
  in f
  end
in let val x = 5
  in g 7
  end
end

let val x = 3
  fun f y = x+y
in let val x = 4
  fun g h = h (h 7)
  in g f
  end
end
2 Parameter Passing (9 points)

In the presence of side-effects, the choice of parameter-passing technique can have non-trivial consequences. To see this, consider the following ML expression:

```ml
val y = ref 1;
fun f x = (!y) + (x + x);
(f (y := (!y)+1; !y)) + (!y);
```

For each of the following parameter passing conventions, indicate the final value computed by this program. (You may assume that + is left-associative.)


   ```ml
   val it = 
   ```

2. [3] call-by-name

   ```ml
   val it = 
   ```


   ```ml
   val it = 
   ```
Convert the function preorder to CPS. Do not convert the functions :: (cons) and @ (append) to CPS; just use them as they are in the CPS version of preorder. If you do not remember what CPS is, your job is to translate the preorder in a tail recursive function.

```
datatype tree =
  Empty
| Bin of int * tree * tree

fun preorder Empty = []
| preorder (Bin(i,t1,t2)) = i :: ((preorder t1) @ (preorder t2))
```
4 Streams in ML - 12 Points

Given the following datatype for streams:

\[
\texttt{datatype stream = S of int * (unit -> stream)}
\]

- Write a function \texttt{genOnes} in ML that generates a stream in which every element is a 1, i.e., generate the stream that represents an infinite sequence of ones.

- Write a function \texttt{addStreams} in ML that takes two streams \texttt{s1} and \texttt{s2} and produces a stream \texttt{s} such that the \texttt{i}th element in \texttt{s} is the sum of the \texttt{i}th elements in \texttt{s1} and \texttt{s2}. 

6
- What would the stream $s$ below represent?

```haskell
fun genS () = S(1, fn () => addStreams (genOnes()) (genS()))
val s = genS()
```
5 Types - 12 Points

Give the ML types of the following functions:

- fun $f(x) = \text{if } x \text{ then } "Bob" \text{ else } "Bill";$

- fun $f(g) = g(g(2))$ ;

- fun $f(x) = \text{if } x = 0 \text{ then } (\text{fn } y \Rightarrow y) \text{ else } h(f(x-1));$

  when $h$ is some polymorphic function of type $(s \rightarrow s) \rightarrow (s \rightarrow s)$

- fun $g(h,x) = \text{fn } y \Rightarrow (h \ x) + (y * 2);$
6 More Types - 12 Points

- Write a function in ML that has the following type: \(\text{int list} \rightarrow \text{int}\).

- Write a function in ML that has the following type:

\[
[\text{'}b\text{'}] \rightarrow [(\text{'}a\text{',}b\text{')] \rightarrow [\text{'}b\text{'}]
\]
- Write a function in ML that has the following type:

\[(a \to b) \to ((a \to b) \to a) \to b\]

- Can you write in ML a function with type: \(a \to b\). Briefly explain your answer.
7 Datatypes - 8 Points

Give ML datatype declarations for any two of the following.

1. A type whose values are the suits of a card deck.

2. A type whose values are "things" where a "thing" is either an integer or a list of "things".

3. A type whose elements are either lists of only integers or lists of only reals.

4. A (polymorphic) type whose values are pairs whose components can be of any type, as long as they are of the same type.
8 Memory Management - 10 Points

This question asks about memory management in the evaluation of the following statically-scoped ML expression.

```ml
val x = 5;
fun f(y) = let val z = [1, 2, 3]
  fun g(w) = w+x+y
  in g end;
val h = let val x=7 in f(3) end;
h(2);
```

1. Draw the run-time structures (with static link, dynamic link, values of local variables, etc.) as they appear after the invocation of `h (2)`.

2. What is the value of the expression?
9 Callcc - 9 Points

What is the result of the following expressions

- 2 + callcc (fn k => 3)

- 2 + callcc (fn k => 3 + (throw k 5) + 7)

- let fun inv n k = if n=0
    then throw k 13
    else (1 div n)
    in 7 + callcc (fn k => 5 + (inv 3 k) + (inv 0 k) + (inv 2 k) + 11)
end
10  More Callcc - 6 Points

Recall that the types for continuations are:

\[
\text{callcc}: (\text{'a cont} \rightarrow \text{'a}) \rightarrow \text{'a}
\]

\[
\text{throw}: \text{'a cont} \rightarrow \text{'a} \rightarrow \text{'b}
\]

Explain why the following expressions do not type check. Can you rewrite the expressions so that they type check? If yes, rewrite the expressions, if not briefly explain why not.

- let val r = fn k => k 6
  in
  ((callcc r) + 3) * 8
  end

- callcc (fn k => k)