Problem 1 [15 points] Provide both the static and dynamic semantics for n-ary products. The syntax of n-ary products is as follows.

\[ \tau ::= \ldots | () | (\tau_1 \cdots \tau_n) \]
\[ e ::= \ldots | (e) | (e_1, \ldots, e_n) \]
\[ \text{split } e_1 \text{ as } () \text{ in } e_2 \]
\[ \text{split } e_1 \text{ as } (x) \text{ in } e_2 \]
\[ \text{split } e_1 \text{ as } (x_1, \ldots, x_n) \text{ in } e_2 \]

You should make sure your rules are clear about the three cases \( n = 0 \), \( n = 1 \), and \( n > 1 \).

Problem 2 [15 points] Provide both the static and dynamic semantics for n-ary sums. The syntax of n-ary sums is as follows.

\[ \tau ::= \ldots | (\tau_1 + \ldots + \tau_n) \]
\[ e ::= \ldots | \text{inj}_{\tau_1 + \ldots + \tau_n}(e) \text{ where } 1 \leq m \leq n \]
\[ \text{case } e \text{ of } \text{inj}_{\tau_1}(x_1 : \tau_1) = e_1 | \ldots | \text{inj}_{\tau_n}(x_n : \tau_n) = e_n \]

You may assume that \( n \geq 2 \) in all your rules.

Problem 3 [10 points] Given the following ML datatype

```
datatype tagged =
    Int of int
  | Bool of bool
  | Pair of tagged * tagged
  | Fun of tagged -> tagged
```

(a) [5 points] Write out a type using the following language of types that is equivalent to the datatype.

```
type vars tv ::= 
\[ \tau ::= \text{int} | \text{unit} | (\tau_1 + \tau_2) | (\tau_1 \cdots \tau_n) \]
\[ (\text{rec } tv \text{ is } \tau) | tv \]
```

(b) [5 points] Given your representation of the type, convert the ML value to an equivalent value based on your answer to part (a). Use the syntax presented in Chapter 19 of Harper to represent your value.

```
Pair(Int 1,Bool true)
```
Problem 4  [10 points] Look in the file tagged.sml. This contains two different implementations of merge sort \texttt{Tagged.merge_sort} and \texttt{Tagged.merge_sort’}.

(a)  [5 points] Explain in one sentence how the two different implementations behave and provide an example input where the behavior of the two functions differ.

(b)  [5 points] For both \texttt{Tagged.merge_sort} and \texttt{Tagged.merge_sort’} explain if we can apply the optimization trick described by Harper in section 24.3. If the optimization can be done provide an optimized version. If the optimization cannot be done explain why.

Problem 5  [25 points] Complete the implementation of \texttt{static-sem.sml}.

Problem 6  [25 points] Complete the implementation of \texttt{m-machine.sml}.