3 Compiler Construction (tgg22)

(a) In the context of the compilation of functions, what is a closure? [2 marks]

(b) The front-end of our Slang compiler eliminates let-bindings by replacing the code

```plaintext
let x e1 in e2 end
```

with the code

```plaintext
(fun x -> e2 end) e1
```

Apply this transformation to the following Slang code.

```plaintext
let f(x) =
    let x1 = e1
    in let x2 = e2
        in e3 end
    in e end
```

[3 marks]

(c) Describe the structure of the Jargon code generated from the Slang in your answer to Part (b). (Don’t worry about getting the syntax exactly right.) [6 marks]

(d) Consider the Jargon code generated in Part (c). Suppose the function f is called with the value v somewhere in the code generated from the expression e. Describe what happens at runtime when f(v) is executed. In particular, describe the closures that exist in the heap and how they are used to evaluate f(v). [4 marks]

(e) Describe a better way of compiling let-bindings such as those associated with x1 and x2 in the code above. Rather than creating closures, the idea is to include these “local variables” in the stack frame for f. Explain in detail how this might be done. [5 marks]