Problem 3 (15 marks)

Part A. [8 marks]

Evaluate the lambda expression below by the interpreter based on context and closure. You are asked to show the context at some point of evaluation. Assume that initial context is CT0 = $\{z \rightarrow 2\}$. Just show your answer to each question. You don't need to show how you derive your answer. This is how the evaluation starts:

```
eval
    ((lambda (f x) (f (f x))) (lambda (u) (* z u)) 5)
in CT0
```

(i) Show the context when `(f (f x))` is evaluated.

Answer: `CT1 = {f -> [(* z u), CT0], x -> 5} U CT0`

(ii) Show the context when `(f x)` is evaluated.

Answer: `CT1` (eval the argument `(f x)` to `f` in the same `CT1`)

(iii) Show the context when `(* z u)` is evaluated.

Answer: `(* z u)` is evaluated twice.

(i)Continuing from eval `(f x)`, evaluate `(* z u)` in `{u -> 5} U CT0`

(ii) Continuing from eval `(f (f x))`, after both `f` and `(f x)` are evaluated, the next is to eval `(* z u)` in `{u -> 10} U CT0`.

(iv) What is the result of this evaluation?

Answer: 20

Part B. [7 marks]

Compile the following lambda function to SECD code.

```
(lambda (x y) ((lambda (z) (+ x y z)) 6))
```

Note that the body of the outside lambda function is an application `((lambda (z) (+ x y z)) 6))`, where the operator + is generic, just like the same operator in lisp. To decompose the overall task, you should first show the SECD code for this expression and denote it by, e.g., `e'`. Then, plug it into the full solution.

A: e' = (NIL LDC 6 CONS LDF (LD (2.1) LD (1.2) LD (1.1) ADD RTN) AP)

Then, the full solution is:

```
(LDF e' || (RTN))
```