Problem 1:

Part A:

```
#Struct: nameless-let-exp
    #(Struct: Const-exp 3)
    #Struct: nameless-let-exp
        #(Struct: Const-exp 2)
        #Struct: nameless-diff-exp
            #(Struct: Nameless-var-exp 1)
            #(Struct: Nameless-var-exp 0)
            #Struct: nameless-diff-exp
                #(Struct: Nameless-var-exp 0)
                #(Struct: Nameless-var-exp 1)
                #Struct: nameless-let-exp
                    #(Struct: Nameless-proc-exp)
                    #(Struct: Nameless-proc-exp)
                    #(Struct: Nameless-proc-exp)
                        #(Struct: Nameless-diff-exp)
                            #(Struct: Nameless-var-exp 0)
                            #(Struct: Nameless-diff-exp)
                                #Struct: Nameless-var-exp 2
                                #Struct: Nameless-var-exp 1
```

Part B:

```
let x = 32 in

    let y = 3 in

        let z = 5 in

            let a = -(x, z)

                let b  = 10 in

                    -(b, -(x, y))
```

Problem 2:

In the first code block, the counter reference is instantiated and saved in g so it will not be instantiated every time g is called. In the second code block, g returns a procedure that has not yet been evaluated and this procedure creates an instance of the counter every time g is called. So the result of the first one is -1 and the second one is 0.