Question 5 (15 points): Short Answer

a. **(5 marks)** Consider the following MIPS code, taken from inside a loop in a procedure. Assume that `$a0, $a1, $a2` are still live after this code, and so will require their own registers. What is the minimum number of registers required to store the intermediate values R0 to R8? Give a mapping from these intermediate values to MIPS `$tX` registers.

```
sll R0, $a0, 2
lw R1, 0($a1)
add R2, R0, R1
lw R3, 0(R2)
sll R4, R3, 2
lw R5, 0($a2)
add R6, R5, R4
lw R7, 0(R6)
addi R8, R7, 1
```

$t0 : R0, R2, R3, R4

$t1 : R1, R5, R6, R7, R8

Minimum = 2

b. **(5 marks)** By convention, registers `$k0` and `$k1` are reserved for use by the kernel. What can happen if these registers are used to store data in a MIPS user program?

If an exception occurs, data in these registers can be overwritten. Therefore, we can't ensure data in $k0/$k1 will persist between instructions.

c. **(5 marks)** Dynamic branch prediction predicts whether a branch is taken or not based on past behaviour. Give one example of a situation where a 2-bit predictor is will be more accurate than a 1-bit predictor, and another example where a 1-bit predictor is more accurate than a 2-bit predictor. For each example, state how many mispredictions each predictor will make.

A 2-bit predictor is more accurate with nested loops, as in the example in class.

A 1-bit predictor will be more accurate in a situation where a branch switches from being always taken to being always not taken, as it will only make 1 misprediction.