1. Exam Question Rework (**Question 4**)

```
.data
str1: .asciiz "MIPS is dead; long live MIPS!"
.text
   lui  $a0, 0x1001      # str1 base address
   ori  $a0, 0x01        # str1 base address
   addiu $a1, $zero, 14   # str1 length
   addiu $t0, $zero, 0
   addiu $t1, $a1, -1
   j    cond
action:
   add  $t3, $a0, $t0
   add  $t4, $a0, $t1
   lb   $t5, 0($t3)
   lb   $t6, 0($t4)
   sb   $t5, 0($t4)
   sb   $t6, 0($t3)
   addiu $t0, $t0, 1
   addiu $t1, $t1, -1
cond:
   slt  $t2, $t0, $t1
   bne  $t2, $zero, action
```

   a. How many instructions would be executed for this example?
      Preloop instructions: 6
      Instructions per iteration: 10 (Includes check condition)
      (Only the last time to break looping) When cond fails: 2
      Calculation:
      10*7 = 70
      6 + 70 + 2 = 78
      Answer: 78

   b. What is the minimum binary size (in bytes) for this sequence of instructions?
      Since there are 16 total lines of instructions, and the question states minimum binary size, 16 is multiplied by 4 because the question asks for the answer in bytes.
      Answer: 16 * 4 = 64

   c. What would the hexadecimal value of $t0 be after the execution of the above code?
      Since the loop increments $t0 during each iteration and stops at 7, the final value in $t0 will be 0x00000007.
      Answer: 0x00000007

   Description of misconception:

In Part A of the problem, I reviewed the program slowly to ensure I wrote my work and counted the iterations correctly. When I put my answer as 76, I wrote it because I thought that after seven iterations, the looping would stop, and the program would exit. This was true, but I forgot to add the last check that will happen before the loop terminates. The answer was supposed to be 78 instead of 76 because I needed to add 2 (the two lines that check whether the condition still holds).

In Part B, I had the misconception that the question asked for the total binary size and decided it would be 4*76, as indicated by my previous answer. However, this was wrong on two counts because my Part A was wrong and was supposed to be 78, and also because Part B is not linked to Part A at all. The correct answer is unrelated to 78 from Part A and should just be 6 + 10 + 2 = 18, and then the 18 is multiplied by 4 to get our answer in bytes.

In Part C of the problem, I knew that the answer was seven iterations, but when I started writing out the answer, I started with 0x and then decided to go with writing 7 in binary instead, which left my answer to be 0x00000111 instead of 0x00000007.