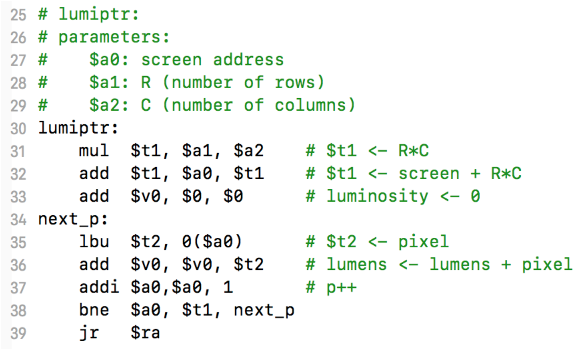
Question 3 (12 points):



```
25 # lumiptr:
26 # parameters:
27 #    $a0: screen address
28 #    $a1: R (number of rows)
29 #    $a2: C (number of columns)
30 lumiptr:
31     mul  $t1, $a1, $a2    # $t1 <- R*C
32     add  $t1, $a0, $t1    # $t1 <- screen + R*C
33     add  $v0, $0, $0      # luminosity <- 0
34 next_p:
35     lbu  $t2, 0($a0)      # $t2 <- pixel
36     add  $v0, $v0, $t2    # lumens <- lumens + pixel
37     addi $a0,$a0, 1       # p++
38     bne  $a0, $t1, next_p
39     jr   $ra
```

Figure 1: The code for `lumiptr`.

1. (**7 points**) What is the binary representation, expressed in hexadecimal, for the `bne` instruction `bne, $a0, $t1, next_p` in line 38 of `lumiptr` in Figure 1?

   The opcode for a bne instruction is `000101`, register `$a0` is register 4 = `00100`, register `$t1` is register 9 = `01001` and the distance between the target of the branch and the instruction after the branch is 4 instructions, therefore the immediate value in the `bne` instruction must be -4 = `1111 1111 1111 1100` Therefore the binary representation of that `beq` instruction is:

   ```
   opcode  rs    rt     immediate
   000101 00100 01001 1111 1111 1111 1100
   0001 0100 1000 1001 1111 1111 1111 1100 = 0x 1489 FFFC
   ```

2. (**5 points**) The code to save/restore register values is omitted from `lumiptr`. Is it necessary for `lumiptr` to save/restore any register values, which ones? Explain your answer.

   There is no need to save/restore any register because `luminosity` does not use `$s` registers or calls any other function.