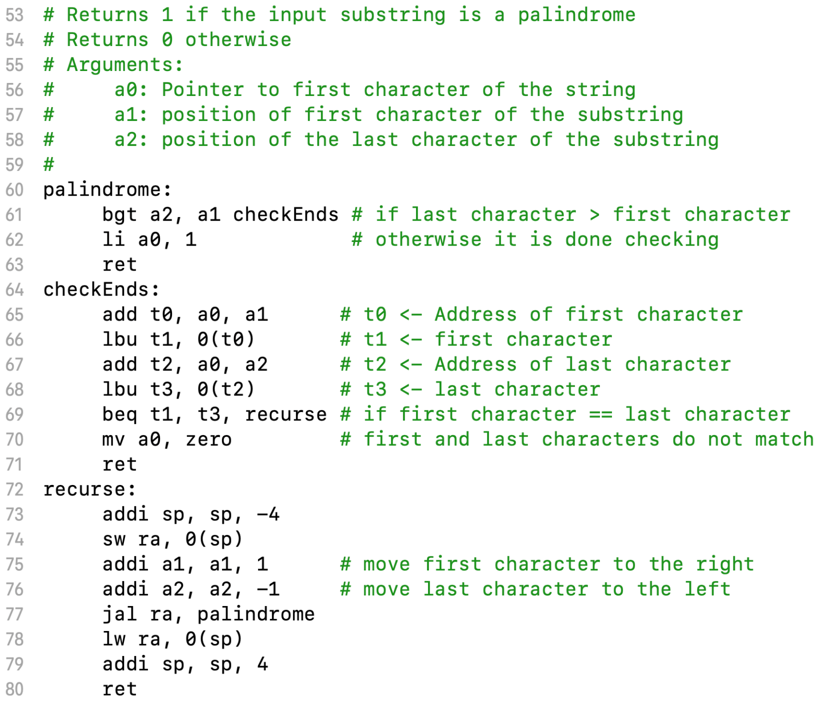
Question 3 (13 points):



```
53 # Returns 1 if the input substring is a palindrome
54 # Returns 0 otherwise
55 # Arguments:
56 #     a0: Pointer to first character of the string
57 #     a1: position of first character of the substring
58 #     a2: position of the last character of the substring
59 #
60 palindrome:
61      bgt a2, a1 checkEnds # if last character > first character
62      li a0, 1             # otherwise it is done checking
63      ret
64 checkEnds:
65      add t0, a0, a1      # t0 <- Address of first character
66      lbu t1, 0(t0)       # t1 <- first character
67      add t2, a0, a2      # t2 <- Address of last character
68      lbu t3, 0(t2)       # t3 <- last character
69      beq t1, t3, recurse # if first character == last character
70      mv a0, zero         # first and last characters do not match
71      ret
72 recurse:
73      addi sp, sp, -4
74      sw ra, 0(sp)
75      addi a1, a1, 1      # move first character to the right
76      addi a2, a2, -1     # move last character to the left
77      jal ra, palindrome
78      lw ra, 0(sp)
79      addi sp, sp, 4
80      ret
```

Figure 1: RISC-V code for function `palindrome`

Figure **??** has the RISC-V assembly code for the recursive function `palindrome`. A string is a palindrome if it reads the same backward or forward. Examples: `deed`, `rotator`, `noon`. Assume that `SP = 0x04A00000` when this function is called with the following parameters:

- Register `a0` contains the address of the null terminated string "detected".
- `a1= 0x00000000`
- `a2 = 0x00000007`

a. (**5 points**) When this recursive function executes, it changes the value of the stack pointer register `SP`. What will be the lowest value written to the register `SP`, expressed in hexadecimal, while executing this recursive `palindrome` with the parameters above?

b. (**3 points**) Assume that the instruction `bgt` in line 61 is at address `0x00400000`, what is the address, expressed in hexadecimal, where the instruction `beq` at line 69 is stored in memory?

c. (**5 points**) What is the binary representation, expressed in hexadecimal, of the instruction `beq` in line 69?