Question 5 (20 points): Write RISC-V assembly for the function `reverseString` that reverses the order of characters in a null terminated string. For example, `reverseString` changes the string `RISC` into the string `CSIR` or the string `ISA` into the string ASI. The resulting string is also null terminated. These are only examples, `reverseString` must work for strings with any number of characters, including an empty string. The parameters for `reverseString` are as follows:

- `a0`: the address of the first character of a null-terminates string.
- `a1`: the number of characters in the string.

`reverseString` has no return value.

The assembly code that you write must follow all the register saving/restoring conventions for RISC-V.

```
39 # reverseString
40 # a0: address of first character of string
41 # a1: non-negative length of string (does not include NULL character)
42 # peudo code:
43 # for(i=0, j=length-1 ; i != j ; i++. j--)
44 #    temp <- S[i]
45 #    S[i] <- S[j]
46 #    S[j] <- temp
47 reverseString:
48     beq  a1, zero, doneString # if string is empty, done
49     mv   t0, zero       # i <- 0
50     addi t1, a1, -1     # j <- length-1
51     bge  t0, t1, doneString # if i>=j, done
52 nextswap:
53     add  t2, a0, t0     # t1 <- address{S[i]}
54     add  t3, a0, t1     # t2 <- address{S[j]}
55     lbu  t4, 0(t2)      # t4 <- S[i]
56     lbu  t5, 0(t3)      # t5 <- S[j]
57     sb   t4, 0(t3)      # S[j] <- S[i]
58     sb   t5, 0(t2)      # S[i] <- S[j]
59     addi t0, t0, 1      # i <- i+1
60     addi t1, t1, -1     # j <- j-1
61     blt  t0, t1, nextswap # if i<j goto nextswap
62 doneString:
63 jalr    zero, ra, 0
```

Figure 1: A solution for `reverseString`.