```
Problem 3: [12 points]
Filename: hw9prob3.asm
AndrewID: jtbell
```

```
; Determine if a string represents a palindrome or not
            .ORG  $FF0      ; Input Data
STRADDR     .DW   STR       ; String address, must be even
STRLEN      .DW   $B        ; Size of the string, in words. Guaranteed not
                            ; to wrap beyond the top of memory
IS_PAL      .DW   $0        ; After completion, 1 if yes, 0 if no
            .ORG  $1000     ; Code segment
                            ; Your code starts here
                            ; At any random spot in memory, there is a string ...
```

```
Line length of 81 (max is 80)
```

```

        LW r1, r0, STRADDR      ; loads the STRADDR to r1=STR
        LW r3, r0, STRLEN       ; loads the STRLEN to r3 r3=$B
        LI  r7, $0001           ; loads 1 as a dummy value for SUB
        ADD r3, r3, r3          ; r3 = 2(strlen)
        ADD r4, r3, r1          ; r4 = straddr + (2strlen)
        ;SUB r4, r4, r7          ; r4 = straddr + (2strlen) - 1
        MV  r5, r1              ; Loads 0 into r5 initially
        MV  r6, r4              ;
        LW  r4, r4, $0          ; LW te addr of


LOOP    SLT r2, r6, r5          ; checking if low < hi
        SUB r7, r7, r2          ; subtracts r7 to r2 to determine if we are
                                ; at the end / condition is true
        BRZ COMPLETELOOP        ; if so complete the loop

        SLT r0, r1, r4          ; checks if the values r1 and r4 are equal
        BRZ NEXT                ; if they are continue looping
        BRA DONE                ; if not continue the loop

NEXT    ADDI r5,r5,$1           ; increases the beginning addr by 1
        SUB  r6,r6,r7           ; decreases the end address by 1
        LW   r1,r5,$0           ; loads the next value for beginning
        LW   r4,r6,$0           ; loads the next value for the end
        BRA  LOOP               ; goes to the nex iteration

COMPLETELOOP LI r7, $1          ;
             SW r0, r7, IS_PAL  ;
             STOP               ;

STR         .DW   $0001
            .DW   $0002
            .DW   $0000
            .DW   $1000
            .DW   $FFFE
            .DW   $EFFF
            .DW   $FFFE
            .DW   $1000
            .DW   $0000
            .DW   $0002
            .DW   $0001
            .DW   $CAFE
            .DW   $DEAD
            .DW   $BEEF

DONE        STOP
```