```
        .data
string: .space 100
copy:   .space 100
prompt: .asciiz "Input String: "
upper:  .asciiz "Frequency of upper case: "
lower:  .asciiz "Frequency of lower case: "
sp:             .asciiz "Frequency of spaces: "
newLine:.asciiz "\n"
pal:    .asciiz "String is a palindrome"
npal:   .asciiz "String is not a palindrome"

        .text
main:
                li $v0, 4       #print prompt
                la $a0,prompt
                syscall
                li $v0, 8       #user input
                la $a0, string
                li $a1, 100
                syscall

                la $a0,string
count:
                #address of string should be in a0
                #t0 - upper case count
                #t1 - lower case count
                #t2 - space count

                li $t0,0
```

```
        li $t1,0
        li $t2,0

        #65             #A
        #90             #Z
        #97             #a
        #122    #z
        #32             #" "
loop:
        lb   $t3,0($a0)
        addi $a0,$a0,1
        beqz $t3,print   #end string
        beq  $t3,32,spc
        bge $t3,97,low
        bge $t3,65,upp
        b loop
spc:
        addi $t2,$t2,1
        b loop
low:
        bge $t3,122,loop
        addi $t1,$t1,1
        b loop
upp:
        bge $t3,90,loop
        addi $t0,$t0,1
        b loop
print:
        la $a0,upper     #print upper prompt
```

```
li $v0,4
syscall
move $a0,$t0    #print count upper case
li $v0,1
syscall
la $a0,newLine  #print new line
li $v0,4
syscall

la $a0,lower      #print lower prompt
li $v0,4
syscall
move $a0,$t1    #print count lower case
li $v0,1
syscall
la $a0,newLine  #print new line
li $v0,4
syscall

la $a0,sp                   #print space prompt
li $v0,4
syscall
move $a0,$t2    #print count space
li $v0,1
syscall
la $a0,newLine  #print new line
li $v0,4
syscall
```

```
                #part 2
                la $a0,string
                jal palindrome
                move $t0,$v0
                beq $t0,1,pals

npals:  #not a palindrome
                la $a0,npal     #print not palindrome
                li $v0,4
                syscall
                b stop
pals:   #is palindrome
                la $a0,pal      #printpalindrome
                li $v0,4
                syscall

stop:
                la $a0,newLine  #print new line
                li $v0,4
                syscall
                la $a0,newLine  #print new line
                li $v0,4
                syscall
                li $v0, 10              #stop
                syscall

palindrome:
                #returns in v0, 0 if false,1 if true
                li $t1,0  #char count
```

```
		li $t3,0  #char
		move $t0,$a0    #spot
		la $a1,copy
		#make a copy with only lowercase letters
		#65		#A
		#90		#Z
		#97		#a
		#122	#z
copyl:
		lb   $t3,0($t0)
		beqz $t3,isPal    #if end string
		addi $t0,$t0,1

		#skip
		bgt $t3,122,ignore  #if > z
		blt $t3,65,ignore   #if < A
		ble $t3,90,uppCase  #if <= Z
		blt $t3,97,ignore		#if < a
		#char is lower case
saveChar:
		sb	$t3,0($a1)	#save char in copy
		addi $t1,$t1,1    #increase char count
		addi $a1,$a1,1
ignore:
		b copyl
uppCase:
		addi $t3,$t3,32 #make char lowercase
		b saveChar
```

```
isPal:
        li $v0,0
        la $t0,copy             #start
        la $t2,copy             #end
        add $t2,$t2,$t1
        addi $t2,$t2,-1
        addi $t1,$t1,1  #offset decrementing before bgtz $t1,loop2
        div $t1,$t1,2   #count should be half because we are looking
                                        #at the front and back


loop2:
        #all chars in copy will be letters and lowercase
        lb $t3,0($t0)#front char
        lb $t4,0($t2)#back char
        addi $t2,$t2,-1
        addi $t0,$t0,1
        addi $t1,$t1,-1 #decrease count
        bne $t3,$t4,back #they aren't equal stop fucntion
        bgtz $t1,loop2  #they are equal continue if still more chars

        li $v0,1 #loop is done so palindrome
        jr $ra  #return
back:   li $v0,0
        jr $ra  #return
```

Jarod Nakamoto
CS 264 Lab 3