# CS/COE 0447 Fall 2009
# Lab 6: Computer Arithmetic
# Solution

```
#Problem 1:

      .data
str:  .space      33    #Extra byte for the null character
buf:  .space      32    #No need for null character because size is fixed

      .text

main: la $a0, str
      li $a1, 33
      li $v0, 8
      syscall
      jal strtobinu
      move $a0, $v0
      li $v0, 1
      syscall
      li $v0, 10
      syscall


strtobinu:
      addi $sp, $sp, -8       #Adjust stack pointer
      sw $s0, 0($sp)          #Save registers
      sw $ra, 4($sp)          #Save return address

      move $s0, $a0           #s0 holds the address of the string
      jal strlen              #Get length of string
      li $t0, 0               #Index into buffer
      addi $t1, $v0, -1       #Index into string
      la $t2, buf             #Address of buffer
_start2:
      beq $t0, 32, _end2      #Loop 32 times
      bge $t0, $v0, _else2    #If index is less than lenght of string, copy
                              #character to buffer
      add $t3, $s0, $t1
      lb $t4, 0($t3)
      j _endif2
_else2:
      li $t4, 48        #Else, copy the character "0" (zero extension)
_endif2:
      add $t3, $t2, $t0
      sb $t4, 0($t3)
      addi $t0, $t0, 1        #Increment index into buffer
      addi $t1, $t1, -1       #Decrement index into string
      j _start2
_end2:
      li $t0, 0         #Index into buffer
      li $v0, 0               #Sum
      li $t6, 1               #Always 1
_start3:
      beq $t0, 32, _end3      #Loop 32 times
      add $t3, $t2, $t0       #Load character
      lb $t4, 0($t3)
      beq $t4, 48, _endif3    #If loaded character is 48, skip
      sllv $t7, $t6, $t0      #Shift 1 index number of times
      add $v0, $v0, $t7       #Add to return value
_endif3:
      addi $t0, $t0, 1        #Increment index
      j _start3
_end3:
      lw $s0, 0($sp)          #Restore registers
      lw $ra, 4($sp)          #Restore return address
      addi $sp, $sp, 8        #Adjust stack pointer
      jr $ra                  #Jump to caller


#This function returns the length of a string that ends in either a null or a
#newline
strlen:
      move $t0, $a0
_start1:
      lb $t1, 0($t0)
      beq $t1, $zero, _end1
      beq $t1, 0x0A, _end1
      addi $t0, $t0, 1
      j _start1
_end1:      sub $v0, $t0, $a0
      jr $ra


#Problem 2:

      .data
str:  .space      33          #Extra byte for the null character
buf:  .space      32          #No need for null character because size is fixed

      .text

main: la $a0, str
      li $a1, 33
      li $v0, 8
      syscall
      jal strtobin
      move $a0, $v0
      li $v0, 1
      syscall
      li $v0, 10
      syscall


strtobin:
      addi $sp, $sp, -8       #Adjust stack pointer
      sw $s0, 0($sp)          #Save registers
      sw $ra, 4($sp)          #Save return address

      move $s0, $a0           #s0 holds the address of the string
      jal strlen              #Get length of string
      li $t0, 0               #Index into buffer
      addi $t1, $v0, -1       #Index into string
      la $t2, buf             #Address of buffer
      li $t4, 48              #If string empty, sign extend with zero
_start2:
      beq $t0, 32, _end2      #Loop 32 times
      bge $t0, $v0, _endif2   #If index is less than lenght of string, copy
                              #character to buffer, else copy the last character
                              #copied
      add $t3, $s0, $t1
      lb $t4, 0($t3)
      j _endif2
_endif2:
      add $t3, $t2, $t0
      sb $t4, 0($t3)
      addi $t0, $t0, 1        #Increment index into buffer
      addi $t1, $t1, -1       #Decrement index into string
      j _start2
_end2:
      li $t0, 0               #Index into buffer
      li $v0, 0               #Sum
      li $t6, 1               #Always 1
_start3:
      beq $t0, 32, _end3      #Loop 32 times
      add $t3, $t2, $t0       #Load character
      lb $t4, 0($t3)
      beq $t4, 48, _endif3    #If loaded character is 48, skip
      sllv $t7, $t6, $t0      #Shift 1 index number of times
      add $v0, $v0, $t7       #Add to return value
_endif3:
      addi $t0, $t0, 1  #Increment index
      j _start3
_end3:
      lw $s0, 0($sp)          #Restore registers
      lw $ra, 4($sp)          #Restore return address
      addi $sp, $sp, 8        #Adjust stack pointer
      jr $ra                  #Jump to caller


#This function returns the length of a string that ends in either a null or a
#newline
strlen:     move $t0, $a0
_start1:    lb $t1, 0($t0)
      beq $t1, $zero, _end1
      beq $t1, 0x0A, _end1
      addi $t0, $t0, 1
      j _start1
_end1:      sub $v0, $t0, $a0
      jr $ra


#Problem 3:

      .data
str:  .space      33    #Extra byte for the null character
buf:  .space      32    #No need for null character because size is fixed

      .text

main: li $v0, 5
      syscall
      move $a0, $v0
      la $a1, str
      jal bintostru
      la $a0, str
      li $v0, 4
      syscall
      li $v0, 10
      syscall


bintostru:
      li $t0, 0               #Bit number
      li $t3, 31              #Length of the string - 1
      la $t5, buf             #Address of buffer
_start1:
      beq $t0, 32, _end1      #Loop 32 times
      srlv $t1, $a0, $t0      #Shift right index number of times
      andi $t1, $t1, 1        #Get rightmost bit
      beq $t1, $zero, _else1  #If righmost bit is zero, jump to else
      li $t2, 49              #Store "1"
      move $t6, $t0           #Keep track of last 1
      j _endif1
_else1:
      li $t2, 48              #Store "0"
_endif1:
      sub $t4, $t3, $t0       #Get index into string
      add $t4, $t4, $t5       #Compute address to store
      sb $t2, 0($t4)          #Store "0" or "1"
      addi $t0, $t0, 1        #Increment index
      j _start1
_end1:                        #Trim leading zeroes
      li $t0, 0               #Index into string
      add $t1, $t6, 1         #Length of string to copy
      sub $t2, $t3, $t6       #Index into buffer
_start2:
      beq $t0, $t1, _end2
      add $t4, $t5, $t2       #Address to load value from
      lb $t3, 0($t4)          #Load value
      add $t4, $a1, $t0       #Address to store value to
      sb $t3, 0($t4)          #Store value
      addi $t0, $t0, 1        #Increment index into string
      addi $t2, $t2, 1        #Increment index into buffer
      j _start2
_end2:
      add $t4, $a1, $t0 #Address of end of string
      sb $zero, 32($a1)       #Put null at end of string
      jr $ra                  #Jump to caller



#Problem 4:

      .data
str:  .space      33    #Extra byte for the null character
buf:  .space      32    #No need for null character because size is fixed

      .text

main: li $v0, 5
```

```
      syscall
      move $a0, $v0
      la $a1, str
      jal bintostr
      la $a0, str
      li $v0, 4
      syscall
      li $v0, 10
      syscall


bintostr:
      li $t0, 0               #Bit number
      li $t3, 31              #Length of the string - 1
_start1:
      beq $t0, 32, _end1      #Loop 32 times
      srlv $t1, $a0, $t0      #Shift right index number of times
      andi $t1, $t1, 1        #Get rightmost bit
      beq $t1, $zero, _else1  #If righmost bit is zero, jump to else
      li $t2, 49              #Store "1"
      j _endif1
_else1:
      li $t2, 48        #Store "0"
_endif1:
      sub $t4, $t3, $t0       #Get index into string
      add $t4, $t4, $a1       #Compute address to store
      sb $t2, 0($t4)          #Store "0" or "1"
      addi $t0, $t0, 1        #Increment index
      j _start1
_end1:
      sb $zero, 32($a1) #Put null at end of string
      jr $ra                  #Jump to caller
```