1.a convertToDec -------------------------------------------------------------------------------------------------------

.data

hexNo: .asciiz "14"

control: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

resultIs: .asciiz "The result is: "

.text

\_\_start:

la $a0, hexNo

jal convertToDec

move $t6, $v0

#Prompt for entering integer count

li $v0, 4

la $a0, resultIs

syscall

#print the result value pritns the values in ascii

li $v0, 1

move $a0, $t6

syscall

li $v0, 10

syscall

convertToDec:

add $t1, $0, $a0

addi $t0, $0, 0 #my counter

# t1 is the address of the header

while:

lb $t3, 0($t1) # I need to take first byte of the string #t1 tooks array[0]

beq $t3, $0 , calculateDecimal

addi $t1, $t1, 1 # increment address to reach other elements of the array

addi $t0, $t0, 1 # my counter $t0 is the length of an array

j while

calculateDecimal:

addi $sp , $sp -4

sw $s1, 0($sp)

#up to now value $t0 = 2

addi $s1, $zero, 0 #my power of hexadecimal

addi $v0, $0, 0 #my su for decimal number

addi $t7, $0, 0 #my su for decimal number

addi $t6, $0, 0

addi $t1, $t1, -1 #to reach the end of the string recover 10

addi $t9, $0, 0

while2:

lb $t3, 0($t1) # I need to take first byte of the string #t1 tooks array[0]

beq $t6, $t0, calculationCompleted

#conversion for ABCDEF

li $t7, 65

beq $t3, $t7, convertA

li $t7, 66

beq $t3, $t7, convertB

li $t7, 67

beq $t3, $t7, convertC

li $t7, 68

beq $t3, $t7, convertD

li $t7, 69

beq $t3, $t7, convertE

li $t7, 70

beq $t3, $t7, convertF

andi $t3, $t3, 0x0F # where $t0 contains the ascii digit

returnCheckPoint:

#li $v0, 1

#move $a0, $t3 #print the number in the string

#syscall

beq $s1, $0, powerZero

returnHere:

bgt $s1, $0, calculatePower

returnHere2:

addi $s1, $s1, 1 # my counter to calculate power of the hex number

addi $t6, $t6, 1 #increase the counter

addi $t1, $t1, -1 # increment address to reach other elements of the array

j while2

calculationCompleted:

add $v0, $0, $t9

lw $s1, 4($sp)

addi $sp, $sp, 4

jr $ra

powerZero:

add $t9, $t9, $t3

#li $v0, 1

#move $a0, $t9 #print the number in the string

#syscall

j returnHere

calculatePower:

addi $t8, $0, 0

while6:

sll $t3, $t3, 4

addi $t8, $t8, 1

beq $s1, $t8, cont

j while6

cont:

add $t9, $t9, $t3

#li $v0, 1

#move $a0, $t9 #print the number in the string

#syscall

j returnHere2

convertA:

addi $t3, $0, 10

j returnCheckPoint

convertB:

addi $t3, $0, 11

j returnCheckPoint

convertC:

addi $t3, $0, 12

j returnCheckPoint

convertD:

addi $t3, $0, 13

j returnCheckPoint

convertE:

addi $t3, $0, 14

j returnCheckPoint

convertF:

addi $t3, $0, 15

j returnCheckPoint

1.b interactWithUser ----------------------------------------------------------------------------------------------------

.data

stringSpace: .space 20

message1: .asciiz "Please enter the string: "

control: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

space1: .asciiz " "

invalidChacterEntered: .asciiz "\nInvalid character has entered, session will be termianted"

sumIs: .asciiz "The result of the conversion is: "

.text

\_\_start:

interactWithUser:

addi $t0, $0, 20 #max size of the string

#Print the message

li $v0, 4

la $a0, message1

syscall

#Take the string

li $v0, 8

la $a0, stringSpace

li $a1, 20

syscall

la $t1, stringSpace # I intend to take head address of the string

add $s3, $zero, $t1 # to have the header

addi $t0, $0, 0 #my counter

#Prints the string

#li $v0, 4

#la $a0, stringSpace

#syscall

# t1 is the address of the header

while:

addi $t6, $0, 0 #my boolean value

lbu $t2, 0($t1) # I need to take first byte of the string

beq $t2, 10, continue2

#print string chars one bye one

addi $t1, $t1, 1 # increment address to reach other elements of the array

addi $t0, $t0, 1 # my counter

#create interval among the numbers

li $v0, 4

la $a0, space1

syscall

#print the values according to index

#li $v0, 1

#move $a0, $t0

#syscall

li $t7, 49

beq $t2, $t7, equalFound

li $t7, 50

beq $t2, $t7, equalFound

li $t7, 51

beq $t2, $t7, equalFound

li $t7, 52

beq $t2, $t7, equalFound

li $t7, 53

beq $t2, $t7, equalFound

li $t7, 54

beq $t2, $t7, equalFound

li $t7, 55

beq $t2, $t7, equalFound

li $t7, 56

beq $t2, $t7, equalFound

li $t7, 57

beq $t2, $t7, equalFound

li $t7, 65

beq $t2, $t7, equalFound

li $t7, 66

beq $t2, $t7, equalFound

li $t7, 67

beq $t2, $t7, equalFound

li $t7, 68

beq $t2, $t7, equalFound

li $t7, 69

beq $t2, $t7, equalFound

li $t7, 70

beq $t2, $t7, equalFound

beq $t6, 0, invalidCharacter

returnHere4:

j while

equalFound:

addi $t6, $0, 1

j returnHere4

invalidCharacter:

#Print the message

li $v0, 4

la $a0, invalidChacterEntered

syscall

li $v0, 10

syscall

continue2:

#print length to be sure

jal calculateDecimal

#Print the message

li $v0, 4

la $a0, sumIs

syscall

li $v0, 1

move $a0, $t9

syscall

li $v0, 10

syscall

calculateDecimal:

addi $sp , $sp -4

sw $s1, 0($sp)

#up to now value $t0 = 2

addi $s1, $zero, 0 #my power of hexadecimal

addi $v0, $0, 0 #my su for decimal number

addi $t7, $0, 0 #my su for decimal number

addi $t6, $0, 0

addi $t1, $t1, -1 #to reach the end of the string recover 10

addi $t9, $0, 0

while2:

lb $t3, 0($t1) # I need to take first byte of the string #t1 tooks array[0]

beq $t6, $t0, calculationCompleted

#conversion for ABCDEF

li $t7, 65

beq $t3, $t7, convertA

li $t7, 66

beq $t3, $t7, convertB

li $t7, 67

beq $t3, $t7, convertC

li $t7, 68

beq $t3, $t7, convertD

li $t7, 69

beq $t3, $t7, convertE

li $t7, 70

beq $t3, $t7, convertF

andi $t3, $t3, 0x0F # where $t0 contains the ascii digit

returnCheckPoint:

#li $v0, 1

#move $a0, $t3 #print the number in the string

#syscall

beq $s1, $0, powerZero

returnHere:

bgt $s1, $0, calculatePower

returnHere2:

addi $s1, $s1, 1 # my counter to calculate power of the hex number

addi $t6, $t6, 1 #increase the counter

addi $t1, $t1, -1 # increment address to reach other elements of the array

j while2

calculationCompleted:

add $v0, $0, $t9

lw $s1, 4($sp)

addi $sp, $sp, 4

jr $ra

powerZero:

add $t9, $t9, $t3

#li $v0, 1

#move $a0, $t9 #print the number in the string

#syscall

j returnHere

calculatePower:

addi $t8, $0, 0

while6:

sll $t3, $t3, 4

addi $t8, $t8, 1

beq $s1, $t8, cont

j while6

cont:

add $t9, $t9, $t3

#li $v0, 1

#move $a0, $t9 #print the number in the string

#syscall

j returnHere2

convertA:

addi $t3, $0, 10

j returnCheckPoint

convertB:

addi $t3, $0, 11

j returnCheckPoint

convertC:

addi $t3, $0, 12

j returnCheckPoint

convertD:

addi $t3, $0, 13

j returnCheckPoint

convertE:

addi $t3, $0, 14

j returnCheckPoint

convertF:

addi $t3, $0, 15

j returnCheckPoint

2. Generating machine instructions

0x10010030 again: add ... # there is an instruction here and meaning is insignificant

0x10010034 add ... # likewise for the other similar cases

0x10010038 add ...

0x1001003C beq $t0, $t1, next

0x10010040 bne $t0, $t1, again

0x10010044 add ...

0x10010048 add ...

0x1001004C next: j again

**Solution:**

* For beq $t0, $t1, next

I type instruction

000100 01001 01010 3

6bit opcode 5bit rs 5 bit rt 16 bit immediate

* For bne $t0, $t1, again

I type instruction

000101 01001 01010 -5

6bit opcode 5bit rs 5 bit rt 16 bit immediate

* For j again

000010 00|0000|0000|0100|0000|0000|1100|last two zero deleted

6 bit opcode 26 bit address

Hexcode 0x0800400C