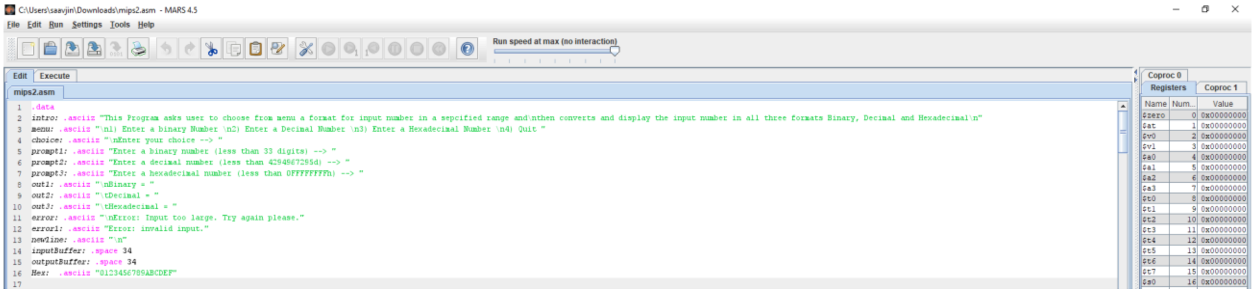
Computer Organization and Architecture

Assembly Project

C3L|CCCS-217
Safanh saadi alzahrani 2210958
Danah saud alsubaie 2210611

The main aim of this project is to create a useful assembly program that can change input numbers into three different types: binary, hexadecimal, and decimal. This project gives users an easy and efficient tool for working with numbers, making it more functional and user-friendly.

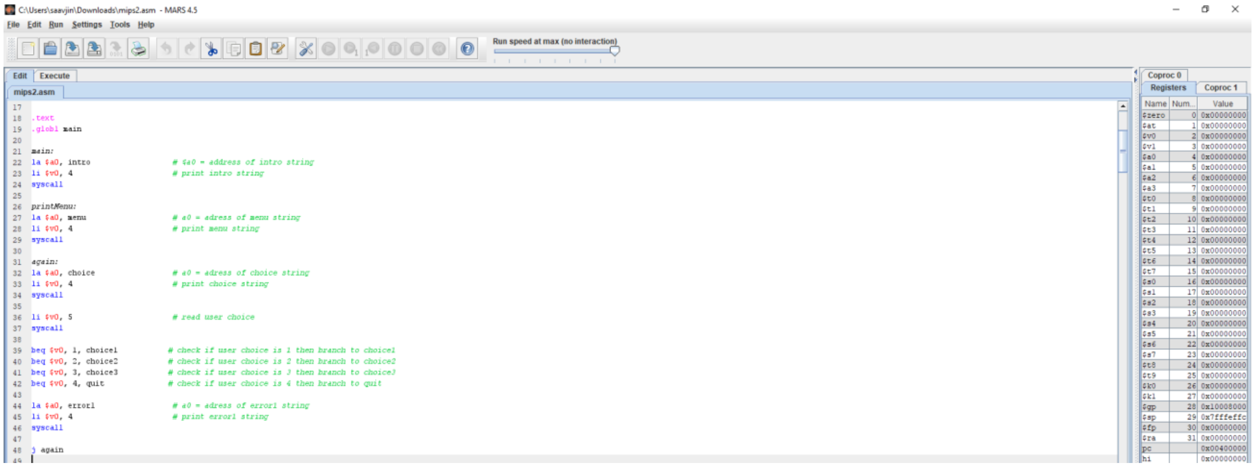
1- This data section declares string and array variables.


C:\Users\saavjin\Downloads\mips2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit
Execute
mips2.asm
Coproc 0
Registers
Coproc 1

```
.data
intro: .asciiz "This Program asks user to choose from menu a format for input number in a sepcified range and\nthen converts and display the input number in all three formats Binary, Decimal and Hexadecimal\n"
menu: .asciiz "\n1) Enter a binary Number \n2) Enter a Decimal Number \n3) Enter a Hexadecimal Number \n4) Quit "
choice: .asciiz "\nEnter your choice --> "
prompt1: .asciiz "Enter a binary number (less than 33 digits) --> "
prompt2: .asciiz "Enter a decimal number (less than 4294967295d) --> "
prompt3: .asciiz "Enter a hexadecimal number (less than 0FFFFFFFFh) --> "
out1: .asciiz "\nBinary = "
out2: .asciiz "\tDecimal = "
out3: .asciiz "\tHexadecimal = "
error: .asciiz "\nError: Input too large. Try again please."
error1: .asciiz "Error: invalid input."
newline: .asciiz "\n"
inputBuffer: .space 34
outputBuffer: .space 34
Hex:  .asciiz "0123456789ABCDEF"
```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000

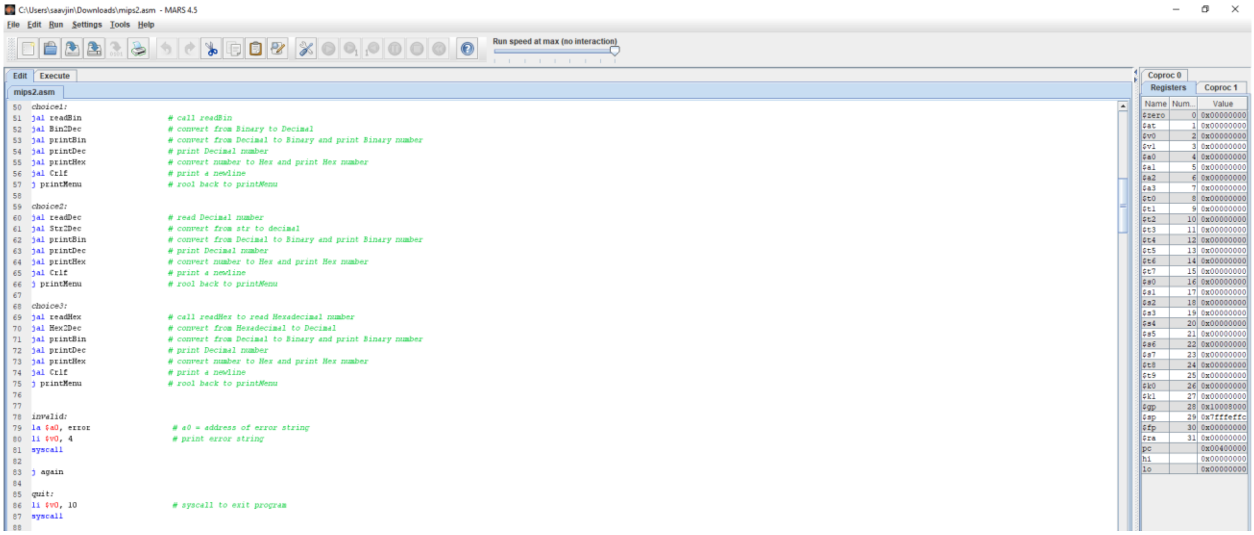
2- the start of the code section, global main indicate that it is the entry point of the program.


C:\Users\saavjin\Downloads\mips2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm
Coproc 0
Registers Coproc 1

```
.text
.globl main

main:
la $a0, intro                   # $a0 = address of intro string
li $v0, 4                       # print intro string
syscall

printMenu:
la $a0, menu                    # a0 = adress of menu string
li $v0, 4                       # print menu string
syscall

again:
la $a0, choice                  # a0 = adress of choice string
li $v0, 4                       # print choice string
syscall

li $v0, 5                       # read user choice
syscall

beq $v0, 1, choice1            # check if user choice is 1 then branch to choice1
beq $v0, 2, choice2            # check if user choice is 2 then branch to choice2
beq $v0, 3, choice3            # check if user choice is 3 then branch to choice3
beq $v0, 4, quit               # check if user choice is 4 then branch to quit

la $a0, error1                  # a0 = adress of error1 string
li $v0, 4                       # print error1 string
syscall

j again
```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000

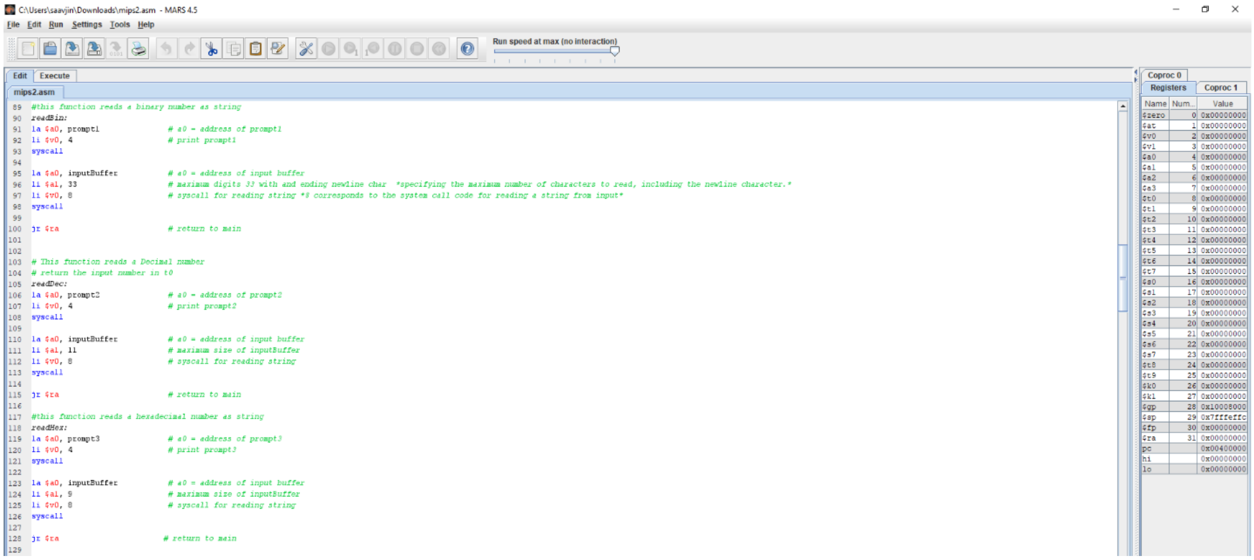
3- user choice section, it has functions inside to convert depending on user choice.


C:\Users\saavjin\Downloads\mips2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm
Coproc 0
Registers Coproc 1

```
choice1:
jal readBin                    # call readBin
jal Bin2Dec                    # convert from Binary to Decimal
jal printBin                   # convert from Decimal to Binary and print Binary number
jal printDec                   # print Decimal number
jal printHex                   # convert number to Hex and print Hex number
jal Crlf                       # print a newline
j printMenu                    # rool back to printMenu

choice2:
jal readDec                    # read Decimal number
jal Str2Dec                    # convert from str to decimal
jal printBin                   # convert from Decimal to Binary and print Binary number
jal printDec                   # print Decimal number
jal printHex                   # convert number to Hex and print Hex number
jal Crlf                       # print a newline
j printMenu                    # rool back to printMenu

choice3:
jal readHex                    # call readHex to read Hexadecimal number
jal Hex2Dec                    # convert from Hexadecimal to Decimal
jal printBin                   # convert from Decimal to Binary and print Binary number
jal printDec                   # print Decimal number
jal printHex                   # convert number to Hex and print Hex number
jal Crlf                       # print a newline
j printMenu                    # rool back to printMenu


invalid:
la $a0, error                   # a0 = address of error string
li $v0, 4                       # print error string
syscall

j again

quit:
li $v0, 10                      # syscall to exit program
syscall
```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

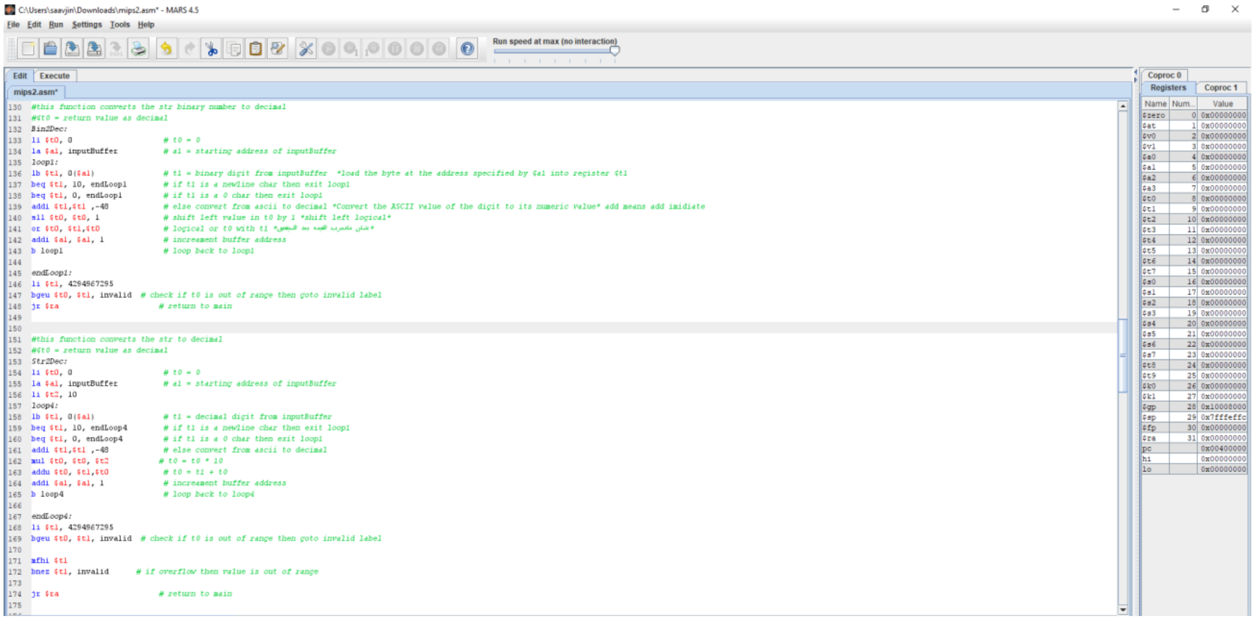
4- These functions first print each prompt message and read the entered number as string then return address to main to perform the conversion methods.


C:\Users\saavjin\Downloads\mips2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm
Coproc 0
Registers Coproc 1

```
#this function reads a binary number as string
readBin:
la $a0, prompt1               # a0 = address of prompt1
li $v0, 4                     # print prompt1
syscall

la $a0, inputBuffer           # a0 = address of input buffer
li $a1, 33                    # maximum digits 33 with and ending newline char  *specifying the maximum number of characters to read, including the newline character.*
li $v0, 8                     # syscall for reading string *8 corresponds to the system call code for reading a string from input*
syscall

jr $ra                        # return to main


# This function reads a Decimal number
# return the input number in t0
readDec:
la $a0, prompt2               # a0 = address of prompt2
li $v0, 4                     # print prompt2
syscall

la $a0, inputBuffer           # a0 = address of input buffer
li $a1, 11                    # maximum size of inputBuffer
li $v0, 8                     # syscall for reading string
syscall

jr $ra                        # return to main

#this function reads a hexadecimal number as string
readHex:
la $a0, prompt3               # a0 = address of prompt3
li $v0, 4                     # print prompt3
syscall

la $a0, inputBuffer           # a0 = address of input buffer
li $a1, 9                     # maximum size of inputBuffer
li $v0, 8                     # syscall for reading string
syscall

jr $ra                       # return to main
```

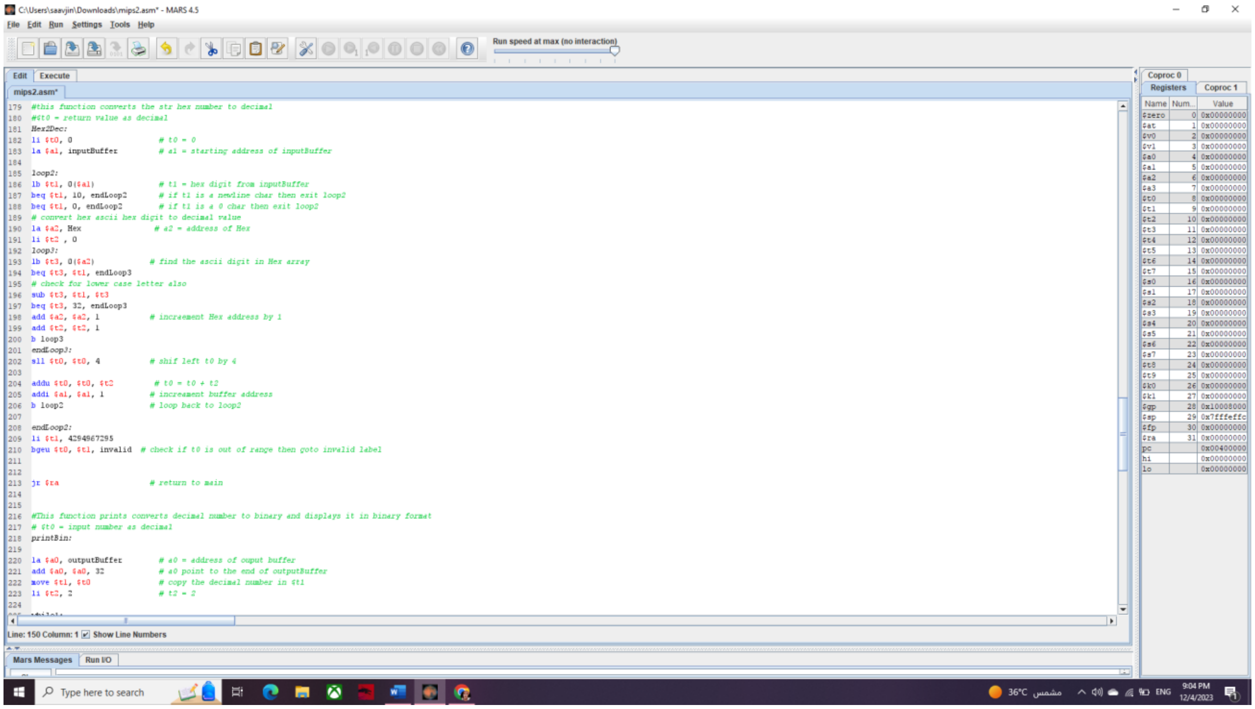
Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

5- Converts functions:


C:\Users\saavjin\Downloads\mips2.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm*
Coproc 0
Registers Coproc 1

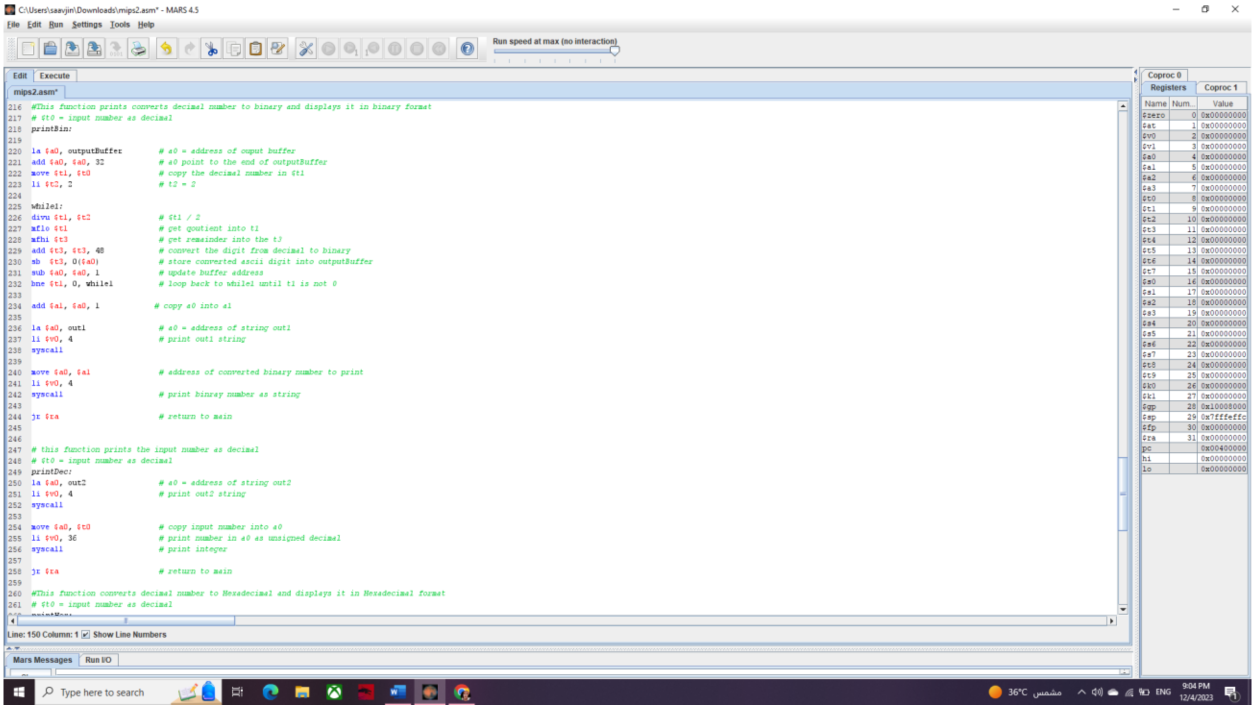
```
#this function converts the str binary number to decimal
#$t0 = return value as decimal
Bin2Dec:
li $t0, 0                    # t0 = 0
la $a1, inputBuffer          # a1 = starting address of inputBuffer
loop1:
lb $t1, 0($a1)               # t1 = binary digit from inputBuffer  *load the byte at the address specified by $a1 into register $t1
beq $t1, 10, endLoop1        # if t1 is a newline char then exit loop1
beq $t1, 0, endLoop1         # if t1 is a 0 char then exit loop1
addi $t1,$t1 ,-48            # else convert from ascii to decimal *Convert the ASCII value of the digit to its numeric value* add means add imidiate
sll $t0, $t0, 1              # shift left value in t0 by 1 *shift left logical*
or $t0, $t1,$t0              # logical or t0 with t1 *عشان ماضرب القيمه بعد الشيفت*
addi $a1, $a1, 1             # increament buffer address
b loop1                      # loop back to loop1

endLoop1:
li $t1, 4294967295
bgeu $t0, $t1, invalid  # check if t0 is out of range then goto invalid label
jr $ra                      # return to main


#this function converts the str to decimal
#$t0 = return value as decimal
Str2Dec:
li $t0, 0                    # t0 = 0
la $a1, inputBuffer          # a1 = starting address of inputBuffer
li $t2, 10
loop4:
lb $t1, 0($a1)               # t1 = decimal digit from inputBuffer
beq $t1, 10, endLoop4        # if t1 is a newline char then exit loop1
beq $t1, 0, endLoop4         # if t1 is a 0 char then exit loop1
addi $t1,$t1 ,-48            # else convert from ascii to decimal
mul $t0, $t0, $t2           # t0 = t0 * 10
addu $t0, $t1,$t0            # t0 = t1 + t0
addi $a1, $a1, 1             # increament buffer address
b loop4                      # loop back to loop4

endLoop4:
li $t1, 4294967295
bgeu $t0, $t1, invalid  # check if t0 is out of range then goto invalid label

mfhi $t1
bnez $t1, invalid      # if overflow then value is out of range

jr $ra                      # return to main
```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000


C:\Users\saavjin\Downloads\mips2.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm*
Line: 150 Column: 1 ☑ Show Line Numbers
Mars Messages Run I/O
Coproc 0 Registers Coproc 1
Type here to search
36°C مشمس ENG
9:04 PM 12/4/2023

```
#this function converts the str hex number to decimal
#$t0 = return value as decimal
Hex2Dec:
li $t0, 0                   # t0 = 0
la $a1, inputBuffer         # a1 = starting address of inputBuffer

loop2:
lb $t1, 0($a1)              # t1 = hex digit from inputBuffer
beq $t1, 10, endLoop2       # if t1 is a newline char then exit loop2
beq $t1, 0, endLoop2        # if t1 is a 0 char then exit loop2
# convert hex ascii hex digit to decimal value
la $a2, Hex                # a2 = address of Hex
li $t2 , 0
loop3:
lb $t3, 0($a2)             # find the ascii digit in Hex array
beq $t3, $t1, endLoop3
# check for lower case letter also
sub $t3, $t1, $t3
beq $t3, 32, endLoop3
add $a2, $a2, 1            # increament Hex address by 1
add $t2, $t2, 1
b loop3
endLoop3:
sll $t0, $t0, 4            # shif left t0 by 4

addu $t0, $t0, $t2          # t0 = t0 + t2
addi $a1, $a1, 1           # increament buffer address
b loop2                    # loop back to loop2

endLoop2:
li $t1, 4294967295
bgeu $t0, $t1, invalid  # check if t0 is out of range then goto invalid label


jr $ra                     # return to main


#This function prints converts decimal number to binary and displays it in binary format
# $t0 = input number as decimal
printBin:

la $a0, outputBuffer        # a0 = address of ouput buffer
add $a0, $a0, 32            # a0 point to the end of outputBuffer
move $t1, $t0               # copy the decimal number in $t1
li $t2, 2                   # t2 = 2

```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000


C:\Users\saavjin\Downloads\mips2.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips2.asm*
Line: 150 Column: 1 ☑ Show Line Numbers
Mars Messages Run I/O
Coproc 0 Registers Coproc 1
Type here to search
36°C مشمس ENG
9:04 PM 12/4/2023

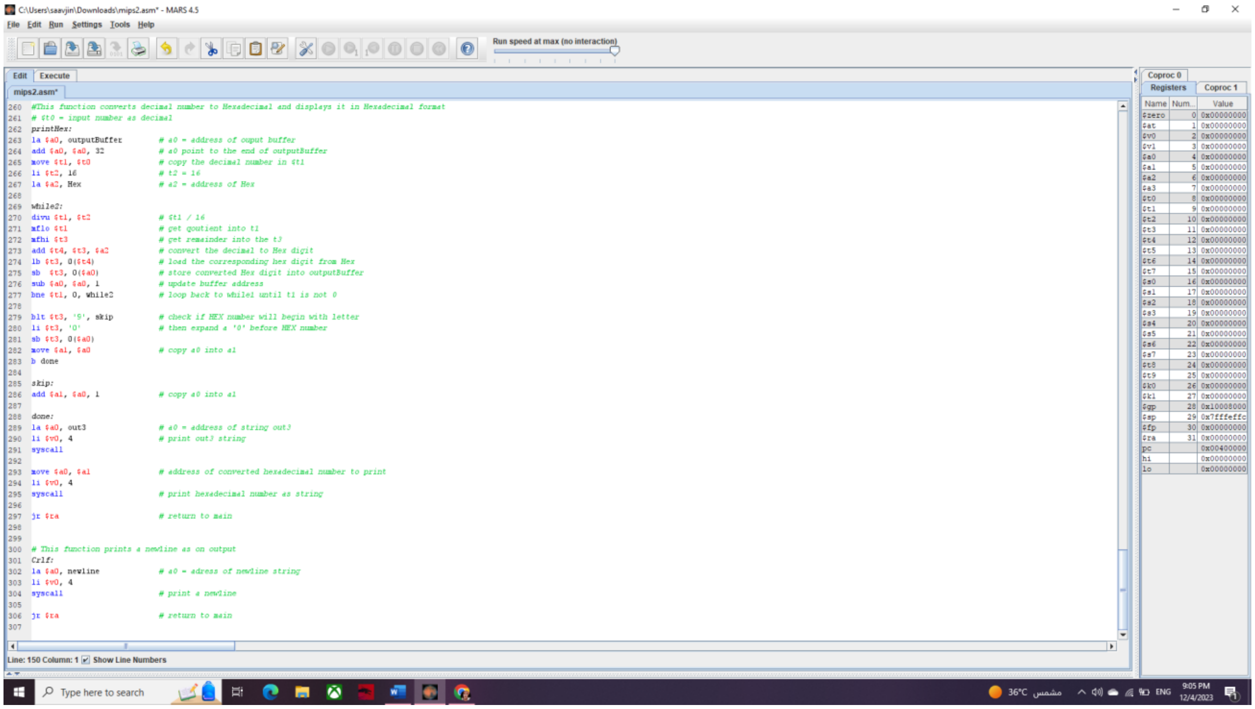
```
#This function prints converts decimal number to binary and displays it in binary format
# $t0 = input number as decimal
printBin:

la $a0, outputBuffer        # a0 = address of ouput buffer
add $a0, $a0, 32            # a0 point to the end of outputBuffer
move $t1, $t0               # copy the decimal number in $t1
li $t2, 2                   # t2 = 2

while1:
divu $t1, $t2               # $t1 / 2
mflo $t1                    # get qoutient into t1
mfhi $t3                    # get remainder into the t3
add $t3, $t3, 48            # convert the digit from decimal to binary
sb  $t3, 0($a0)             # store converted ascii digit into outputBuffer
sub $a0, $a0, 1             # update buffer address
bne $t1, 0, while1          # loop back to while1 until t1 is not 0

add $a1, $a0, 1            # copy a0 into a1

la $a0, out1                # a0 = address of string out1
li $v0, 4                   # print out1 string
syscall

move $a0, $a1               # address of converted binary number to print
li $v0, 4
syscall                     # print binray number as string

jr $ra                      # return to main


# this function prints the input number as decimal
# $t0 = input number as decimal
printDec:
la $a0, out2                # a0 = address of string out2
li $v0, 4                   # print out2 string
syscall

move $a0, $t0               # copy input number into a0
li $v0, 36                  # print number in a0 as unsigned decimal
syscall                     # print integer

jr $ra                      # return to main

#This function converts decimal number to Hexadecimal and displays it in Hexadecimal format
# $t0 = input number as decimal
```

Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

C3L|CCCS-217

Safanh saadi alzahrani 2210958

Danah saud alsubaie 2210611



C:\Users\saavjin\Downloads\mips2.asm* - MARS 4.5

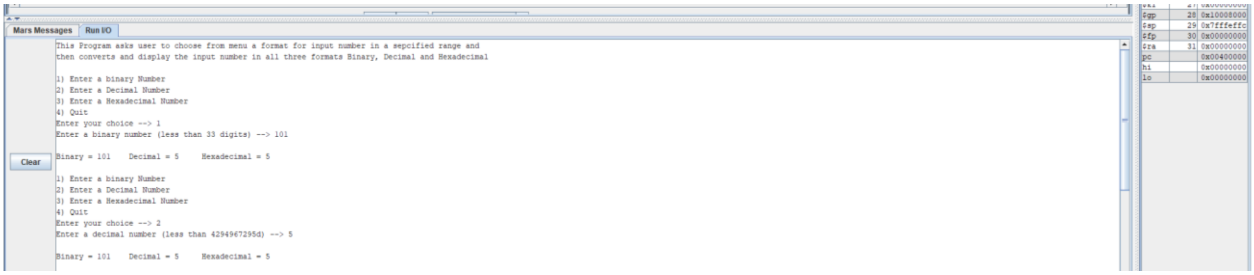
```
#This function converts decimal number to Hexadecimal and displays it in Hexadecimal format
# $t0 = input number as decimal
printHex:
la $a0, outputBuffer        # a0 = address of ouput buffer
add $a0, $a0, 32            # a0 point to the end of outputBuffer
move $t1, $t0               # copy the decimal number in $t1
li $t2, 16                  # t2 = 16
la $a2, Hex                 # a2 = address of Hex

while2:
divu $t1, $t2               # $t1 / 16
mflo $t1                    # get qoutient into t1
mfhi $t3                    # get remainder into the t3
add $t4, $t3, $a2           # convert the decimal to Hex digit
lb $t3, 0($t4)              # load the corresponding hex digit from Hex
sb  $t3, 0($a0)             # store converted Hex digit into outputBuffer
sub $a0, $a0, 1             # update buffer address
bne $t1, 0, while2          # loop back to while1 until t1 is not 0

blt $t3, '9', skip          # check if HEX number will begin with letter
li $t3, '0'                 # then expand a '0' before HEX number
sb $t3, 0($a0)
move $a1, $a0               # copy a0 into a1
b done

skip:
add $a1, $a0, 1             # copy a0 into a1

done:
la $a0, out3                # a0 = address of string out3
li $v0, 4                   # print out3 string
syscall

move $a0, $a1               # address of converted hexadecimal number to print
li $v0, 4
syscall                     # print hexadecimal number as string

jr $ra                      # return to main


# This function prints a newline as on output
Crlf:
la $a0, newline             # a0 = adress of newline string
li $v0, 4
syscall                     # print a newline

jr $ra                      # return to main
```

Line: 150 Column: 1 ☑ Show Line Numbers

Coproc 0 | Registers | Coproc 1

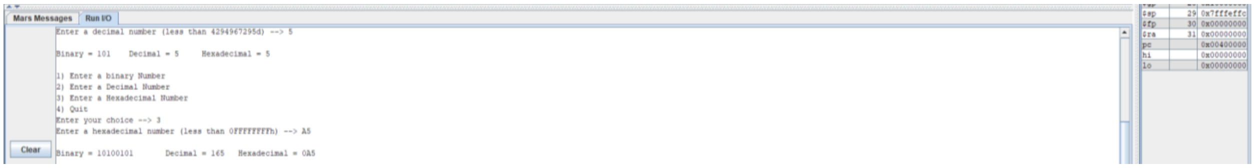
Name	Num...	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Output:



```
This Program asks user to choose from menu a format for input number in a sepcified range and
then converts and display the input number in all three formats Binary, Decimal and Hexadecimal

1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Enter your choice --> 1
Enter a binary number (less than 33 digits) --> 101

Binary = 101    Decimal = 5     Hexadecimal = 5

1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Enter your choice --> 2
Enter a decimal number (less than 4294967295d) --> 5

Binary = 101    Decimal = 5     Hexadecimal = 5
```

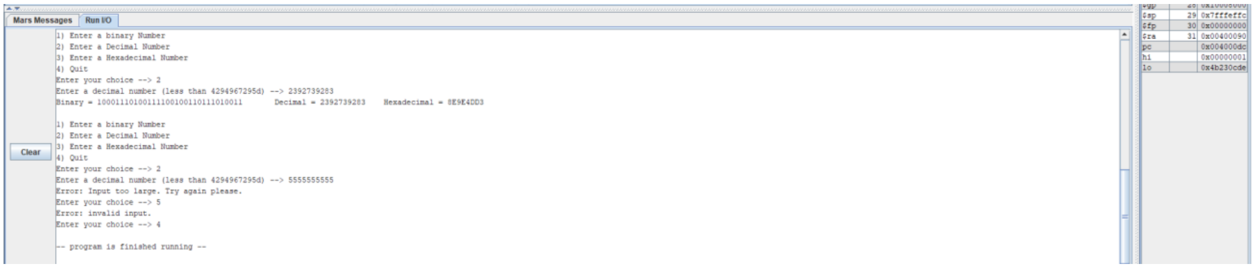
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000



```
Enter a decimal number (less than 4294967295d) --> 5

Binary = 101    Decimal = 5     Hexadecimal = 5

1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Enter your choice --> 3
Enter a hexadecimal number (less than 0FFFFFFFFh) --> A5

Binary = 10100101       Decimal = 165   Hexadecimal = 0A5
```

$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

C3L|CCCS-217
Safanh saadi alzahrani 2210958
Danah saud alsubaie 2210611



```
1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Enter your choice --> 2
Enter a decimal number (less than 4294967295d) --> 2392739283
Binary = 10001110100111100100110111010011       Decimal = 2392739283    Hexadecimal = 8E9E4DD3

1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Enter your choice --> 2
Enter a decimal number (less than 4294967295d) --> 5555555555
Error: Input too large. Try again please.
Enter your choice --> 5
Error: invalid input.
Enter your choice --> 4

-- program is finished running --
```

$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00400090
pc		0x004000dc
hi		0x00000001
lo		0x4b230cde