

Computer Organization and Architecture

Assembly Project

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.



```

1  .data
2  intro: .asciiz "This Program asks user to choose from menu a format for input number in a specified range and then converts and display the input number in all three formats Binary, Decimal and Hexadecimal"
3  menu: .asciiz "\n1) Enter a binary Number \n2) Enter a Decimal Number \n3) Enter a Hexadecimal Number \n4) Quit "
4  choice: .asciiz "Enter your choice --> "
5  prompt1: .asciiz "Enter a binary number (less than 33 digits) --> "
6  prompt2: .asciiz "Enter a decimal number (less than 4294967295d) --> "
7  prompt3: .asciiz "Enter a hexadecimal number (less than 0FFFFFFFh) --> "
8  out1: .asciiz "\nBinary = "
9  out2: .asciiz "\nDecimal = "
10 out3: .asciiz "\nHexadecimal = "
11 error: .asciiz "\nError: Input too large. Try again please."
12 error1: .asciiz "\nError: invalid input."
13 newline: .asciiz "\n"
14 inputBuffer: .space 34
15 outputBuffer: .space 34
16 hex: .asciiz "0123456789ABCDEF"
17

```

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

C:\Users\saadi\Downloads\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

```

17
18 .text
19 global main
20
21 main:
22     la $a0, intro      # $a0 = address of intro string
23     li $v0, 4           # print intro string
24     syscall
25
26     printMenu:
27     la $a0, menu       # $a0 = address of menu string
28     li $v0, 4           # print menu string
29     syscall
30
31     again:
32     la $a0, choice     # $a0 = address of choice string
33     li $v0, 4           # print choice string
34     syscall
35
36     li $v0, 5           # read user choice
37     syscall
38
39     beq $v0, 1, choice1 # check if user choice is 1 then branch to choice1
40     beq $v0, 2, choice2 # check if user choice is 2 then branch to choice2
41     beq $v0, 3, choice3 # check if user choice is 3 then branch to choice3
42     beq $v0, 4, quit    # check if user choice is 4 then branch to quit
43
44     la $a0, error1     # $a0 = address of error1 string
45     li $v0, 4           # print error1 string
46     syscall
47
48     j again
49
50

```

Coprocs		
Registers	Coprocc 0	Coprocc 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
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x00000000
\$fp	29	0xffffffff
\$fpc	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00000000
\$hi		0x00000000

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

C:\Users\saadi\Downloads\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

```

50 choice1:
51     jal readBin        # call readBin
52     jal Bin2Dec        # convert from Binary to Decimal
53     jal printBin       # convert from Decimal to Binary and print Binary number
54     jal printDec       # print Decimal number
55     jal printHex       # convert number to Hex and print Hex number
56     jal CrLf           # print a newline
57     j printMenu        # roll back to printMenu
58
59 choice2:
60     jal readDec        # read Decimal number
61     jal Str2Dec        # convert from str to decimal
62     jal printBin       # convert from Decimal to Binary and print Binary number
63     jal printDec       # print Decimal number
64     jal printHex       # convert number to Hex and print Hex number
65     jal CrLf           # print a newline
66     j printMenu        # roll back to printMenu
67
68 choice3:
69     jal readHex        # call readHex to read Hexadecimal number
70     jal Hex2Dec        # convert from Hexadecimal to Decimal
71     jal printBin       # convert from Decimal to Binary and print Binary number
72     jal printDec       # print Decimal number
73     jal printHex       # convert number to Hex and print Hex number
74     jal CrLf           # print a newline
75     j printMenu        # roll back to printMenu
76
77
78 invalid:
79     la $a0, error      # $a0 = address of error string
80     li $v0, 4           # print error string
81     syscall
82
83     j again
84
85     quit:
86     li $v0, 10         # syscall to exit program
87     syscall
88

```

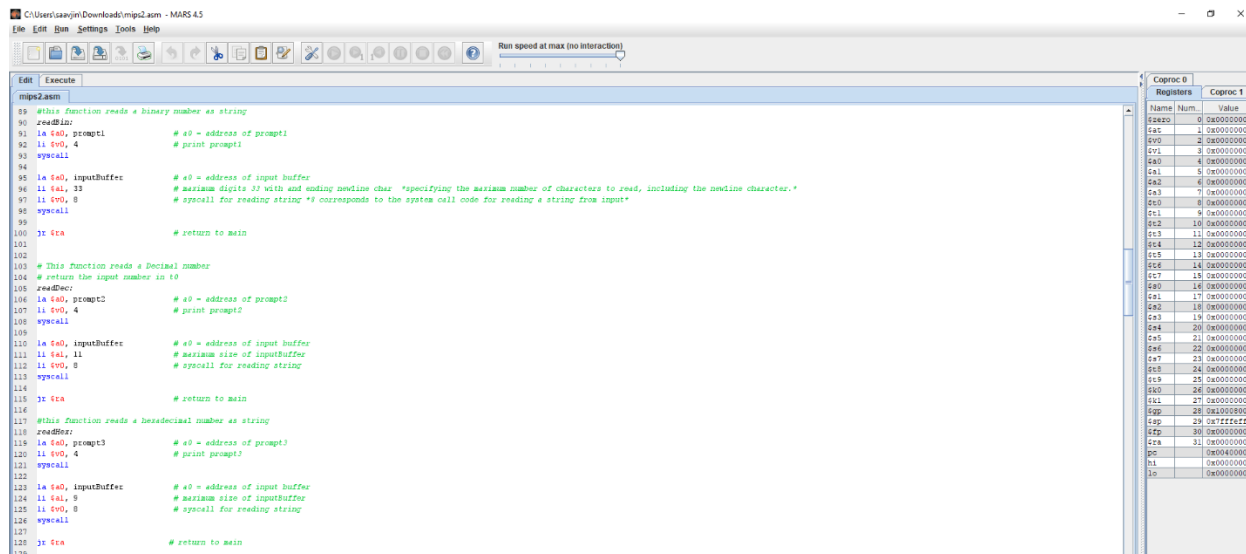
Coprocs		
Registers	Coprocc 0	Coprocc 1
Name	Num	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	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
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x00000000
\$fp	29	0xffffffff
\$fpc	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00000000
\$hi		0x00000000
\$lo		0x00000000

C3L|CCCS-217

Safanh saadi alzahrani 2210958

Danah saud alsubaie 2210611

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.

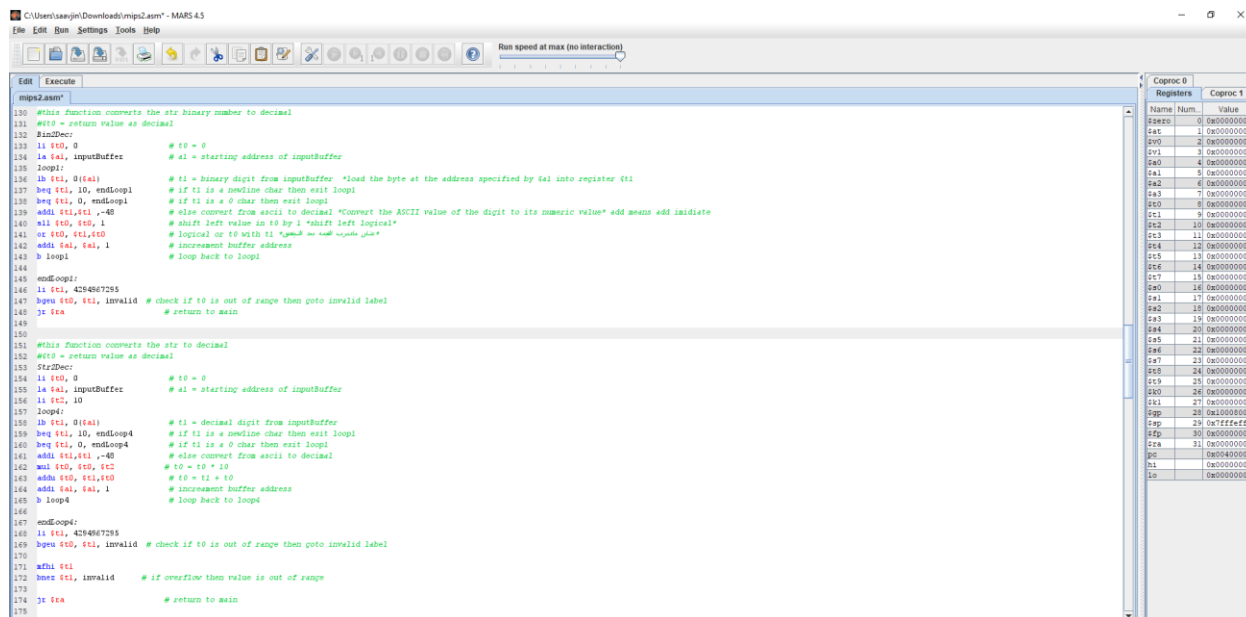


```

185 #this function reads a binary number as string
186 readBin:
187 la $a0, prompt1      # $a0 = address of prompt1
188 li $v0, 4             # print prompt1
189 syscall
190
191 la $a0, inputBuffer   # $a0 = address of input buffer
192 li $a1, 33            # maximum digits 33 with end ending newline char "specifying the maximum number of characters to read, including the newline character."
193 li $v0, 8             # syscall for reading string "l" corresponds to the system call code for reading a string from input
194 syscall
195
196 jr $ra               # return to main
197
198 # This function reads a Decimal number
199 # return the input number in $t0
200 readDec:
201 la $a0, prompt2      # $a0 = address of prompt2
202 li $v0, 4             # print prompt2
203 syscall
204
205 la $a0, inputBuffer   # $a0 = address of input buffer
206 li $a1, 11            # maximum size of input buffer
207 li $v0, 8             # syscall for reading string
208 syscall
209
210 jr $ra               # return to main
211
212 #this function reads a hexadecimal number as string
213 readHex:
214 la $a0, prompt3      # $a0 = address of prompt3
215 li $v0, 4             # print prompt3
216 syscall
217
218 la $a0, inputBuffer   # $a0 = address of input buffer
219 li $a1, 9             # maximum size of input buffer
220 li $v0, 8             # syscall for reading string
221 syscall
222
223 jr $ra               # return to main
224

```

5- Converts functions:



```

225 #this function converts the str binary number to decimal
226 # $t0 = return value as decimal
227 Bin2Dec:
228 li $t0, 0             # $t0 = 0
229 la $a1, inputBuffer   # $a1 = starting address of input buffer
230 loop1:
231 lb $t1, 0($a1)        # $t1 = binary digit from input buffer "load the byte at the address specified by $a1 into register $t1"
232 beq $t1, 10, endLoop1 # if $t1 is a newline char then exit loop1
233 beq $t1, 0, endLoop1  # if $t1 is a 0 char then exit loop1
234 addi $t1, $t1, -48     # else convert from ascii to decimal "Convert the ASCII value of the digit to its numeric value" add means add immediate
235 sll $t0, $t0, 1        # shift left value in $t0 by 1 "shift left logical"
236 add $t0, $t0, $t1      # logical or to with $t1 "add means add immediate"
237 addi $a1, $a1, 1      # increment buffer address
238 b loop1               # loop back to loop1
239
240 endLoop1:
241 li $t1, 4294967295    # check if $t0 is out of range then goto invalid label
242 bgeu $t0, $t1, invalid # return to main
243 jr $ra
244
245 #this function converts the str to decimal
246 # $t0 = return value as decimal
247 Str2Dec:
248 li $t0, 0             # $t0 = 0
249 la $a1, inputBuffer   # $a1 = starting address of input buffer
250 li $t2, 10
251 loop1:
252 lb $t1, 0($a1)        # $t1 = decimal digit from input buffer
253 beq $t1, 10, endLoop4 # if $t1 is a newline char then exit loop1
254 beq $t1, 0, endLoop4  # if $t1 is a 0 char then exit loop1
255 addi $t1, $t1, -48     # else convert from ascii to decimal
256 mul $t0, $t0, $t2      # $t0 = $t0 * 10
257 addu $t0, $t0, $t1      # $t0 = $t0 + $t1
258 addi $a1, $a1, 1      # increment buffer address
259 b loop1               # loop back to loop1
260
261 endLoop4:
262 li $t1, 4294967295    # check if $t0 is out of range then goto invalid label
263 bgeu $t0, $t1, invalid
264
265 mfhi $t1               # if overflow then value is out of range
266 bnez $t1, invalid
267 jr $ra               # return to main
268

```

C3L|CCCS-217

Safanh saadi alzahrani 2210958

Danah saud alsubaie 2210611

C:\Users\saadi\Downloads\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

```

179 #this function converts the str hex number to decimal
180 #t0 = return value as decimal
181 #endDec:
182 li t0, 0          # t0 = 0
183 la t1, inputBuffer # a1 = starting address of inputBuffer
184
185 loop2:
186 lb t1, 0(t1)       # t1 = hex digit from inputBuffer
187 beq t1, 10, endLoop2 # if t1 is a newline char then exit loop2
188 beq t1, 0, endLoop2 # if t1 is a 0 char then exit loop2
189 # convert hex ascii hex digit to decimal value
190 la t2, Hex         # a2 = address of Hex
191 li t2, 0
192 loop3:
193 lb t2, 0(t2)       # find the ascii digit in Hex array
194 beq t2, t1, endLoop3 # check for lower case letter also
195 # check for lower case letter also
196 sub t2, t1, t2      # t2 = t1 - t2
197 beq t2, 32, endLoop3 # increment Hex address by 1
198 add t2, t2, 1
199 add t2, t2, 1
200 b loop3
201 endLoop3:
202 sll t0, t0, 4       # shift left t0 by 4
203
204 addu t0, t0, t2      # t0 = t0 + t2
205 addi t1, t1, 1       # increment buffer address
206 b loop2
207
208 endLoop2:
209 li t1, 4294967295   # check if t0 is out of range then goto invalid label
210 bgtu t0, t1, invalid # check if t0 is out of range then goto invalid label
211
212 jx t0a              # return to main
213
214
215
216 #this function prints converts decimal number to binary and displays it in binary format
217 # t0 = input number as decimal
218 #printBin:
219
220 la t0, outputBuffer # a0 = address of output buffer
221 add t0, t0, 32        # a0 point to the end of outputBuffer
222 move t1, t0          # copy the decimal number in t1
223 li t2, 2             # t2 = 2
224
225 while1:
226 divu t1, t2          # t1 / 2
227 mflo t1              # get quotient into t1
228 mfhi t3              # get remainder into the t3
229 add t3, t3, 48        # convert the digit from decimal to binary
230 sb t3, 0(t0)          # store converted ascii digit into outputBuffer
231 sub t0, t0, 1         # update buffer address
232 bne t1, 0, while1    # loop back to while1 until t1 is not 0
233
234 addi t1, t0, 1        # copy a0 into a1
235
236 la t0, out1          # a0 = address of string out1
237 li t0, 4             # print out1 string
238 syscall
239
240 move t0, t1           # address of converted binary number to print
241 li t0, 4             # print t0 string
242 syscall
243
244 jx t0a              # return to main
245
246
247 # this function prints the input number as decimal
248 # t0 = input number as decimal
249 #printDec:
250 la t0, out2          # a0 = address of string out2
251 li t0, 4             # print out2 string
252 syscall
253
254 move t0, t0          # copy input number into a0
255 li t0, 36            # print number in a0 as unsigned decimal
256 syscall
257
258 jx t0a              # return to main
259
260 #this function converts decimal number to Hexadecimal and displays it in Hexadecimal format
261 # t0 = input number as decimal

```

Line: 150 Column: 1 Show Line Numbers

Mars Messages Run IO

Type here to search

36°C مشمس 9:04 PM 12/4/2023

Copro 0			Copro 1		
Name	Num	Value	Name	Num	Value
\$zero	0	0x00000000	\$t0	0	0x00000000
\$at	1	0x00000000	\$t1	1	0x00000000
\$v0	2	0x00000000	\$t2	2	0x00000000
\$t1	3	0x00000000	\$t3	3	0x00000000
\$a0	4	0x00000000	\$t4	4	0x00000000
\$a1	5	0x00000000	\$t5	5	0x00000000
\$a2	6	0x00000000	\$t6	6	0x00000000
\$a3	7	0x00000000	\$t7	7	0x00000000
\$t0	8	0x00000000	\$t8	8	0x00000000
\$t1	9	0x00000000	\$t9	9	0x00000000
\$t2	10	0x00000000	\$s0	10	0x00000000
\$t3	11	0x00000000	\$s1	11	0x00000000
\$t4	12	0x00000000	\$s2	12	0x00000000
\$t5	13	0x00000000	\$s3	13	0x00000000
\$t6	14	0x00000000	\$s4	14	0x00000000
\$t7	15	0x00000000	\$s5	15	0x00000000
\$t8	16	0x00000000	\$s6	16	0x00000000
\$t9	17	0x00000000	\$s7	17	0x00000000
\$s0	18	0x00000000	\$s8	18	0x00000000
\$s1	19	0x00000000	\$s9	19	0x00000000
\$s2	20	0x00000000	\$t0	20	0x00000000
\$s3	21	0x00000000	\$t1	21	0x00000000
\$s4	22	0x00000000	\$t2	22	0x00000000
\$s5	23	0x00000000	\$t3	23	0x00000000
\$s6	24	0x00000000	\$t4	24	0x00000000
\$s7	25	0x00000000	\$t5	25	0x00000000
\$s8	26	0x00000000	\$t6	26	0x00000000
\$s9	27	0x00000000	\$t7	27	0x00000000
\$t0	28	0x00000000	\$t8	28	0x00000000
\$t1	29	0x00000000	\$t9	29	0x00000000
\$t2	30	0x00000000	\$s0	30	0x00000000
\$t3	31	0x00000000	\$s1	31	0x00000000
\$t4	32	0x00000000	\$s2	32	0x00000000
\$t5	33	0x00000000	\$s3	33	0x00000000
\$t6	34	0x00000000	\$s4	34	0x00000000
\$t7	35	0x00000000	\$s5	35	0x00000000
\$t8	36	0x00000000	\$s6	36	0x00000000
\$t9	37	0x00000000	\$s7	37	0x00000000
\$s0	38	0x00000000	\$s8	38	0x00000000
\$s1	39	0x00000000	\$s9	39	0x00000000
\$s2	40	0x00000000	\$t0	40	0x00000000
\$s3	41	0x00000000	\$t1	41	0x00000000
\$s4	42	0x00000000	\$t2	42	0x00000000
\$s5	43	0x00000000	\$t3	43	0x00000000
\$s6	44	0x00000000	\$t4	44	0x00000000
\$s7	45	0x00000000	\$t5	45	0x00000000
\$s8	46	0x00000000	\$t6	46	0x00000000
\$s9	47	0x00000000	\$t7	47	0x00000000
\$t0	48	0x00000000	\$t8	48	0x00000000
\$t1	49	0x00000000	\$t9	49	0x00000000
\$t2	50	0x00000000	\$s0	50	0x00000000
\$t3	51	0x00000000	\$s1	51	0x00000000
\$t4	52	0x00000000	\$s2	52	0x00000000
\$t5	53	0x00000000	\$s3	53	0x00000000
\$t6	54	0x00000000	\$s4	54	0x00000000
\$t7	55	0x00000000	\$s5	55	0x00000000
\$t8	56	0x00000000	\$s6	56	0x00000000
\$t9	57	0x00000000	\$s7	57	0x00000000
\$s0	58	0x00000000	\$s8	58	0x00000000
\$s1	59	0x00000000	\$s9	59	0x00000000
\$s2	60	0x00000000	\$t0	60	0x00000000
\$s3	61	0x00000000	\$t1	61	0x00000000
\$s4	62	0x00000000	\$t2	62	0x00000000
\$s5	63	0x00000000	\$t3	63	0x00000000
\$s6	64	0x00000000	\$t4	64	0x00000000
\$s7	65	0x00000000	\$t5	65	0x00000000
\$s8	66	0x00000000	\$t6	66	0x00000000
\$s9	67	0x00000000	\$t7	67	0x00000000
\$t0	68	0x00000000	\$t8	68	0x00000000
\$t1	69	0x00000000	\$t9	69	0x00000000
\$t2	70	0x00000000	\$s0	70	0x00000000
\$t3	71	0x00000000	\$s1	71	0x00000000
\$t4	72	0x00000000	\$s2	72	0x00000000
\$t5	73	0x00000000	\$s3	73	0x00000000
\$t6	74	0x00000000	\$s4	74	0x00000000
\$t7	75	0x00000000	\$s5	75	0x00000000
\$t8	76	0x00000000	\$s6	76	0x00000000
\$t9	77	0x00000000	\$s7	77	0x00000000
\$s0	78	0x00000000	\$s8	78	0x00000000
\$s1	79	0x00000000	\$s9	79	0x00000000
\$s2	80	0x00000000	\$t0	80	0x00000000
\$s3	81	0x00000000	\$t1	81	0x00000000
\$s4	82	0x00000000	\$t2	82	0x00000000
\$s5	83	0x00000000	\$t3	83	0x00000000
\$s6	84	0x00000000	\$t4	84	0x00000000
\$s7	85	0x00000000	\$t5	85	0x00000000
\$s8	86	0x00000000	\$t6	86	0x00000000
\$s9	87	0x00000000	\$t7	87	0x00000000
\$t0	88	0x00000000	\$t8	88	0x00000000
\$t1	89	0x00000000	\$t9	89	0x00000000
\$t2	90	0x00000000	\$s0	90	0x00000000
\$t3	91	0x00000000	\$s1	91	0x00000000
\$t4	92	0x00000000	\$s2	92	0x00000000
\$t5	93	0x00000000	\$s3	93	0x00000000
\$t6	94	0x00000000	\$s4	94	0x00000000
\$t7	95	0x00000000	\$s5	95	0x00000000
\$t8	96	0x00000000	\$s6	96	0x00000000
\$t9	97	0x00000000	\$s7	97	0x00000000
\$s0	98	0x00000000	\$s8	98	0x00000000
\$s1	99	0x00000000	\$s9	99	0x00000000

C:\Users\saadi\Downloads\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

```

216 #this function prints converts decimal number to binary and displays it in binary format
217 # t0 = input number as decimal
218 #printBin:
219
220 la t0, outputBuffer # a0 = address of output buffer
221 add t0, t0, 32        # a0 point to the end of outputBuffer
222 move t1, t0          # copy the decimal number in t1
223 li t2, 2             # t2 = 2
224
225 while1:
226 divu t1, t2          # t1 / 2
227 mflo t1              # get quotient into t1
228 mfhi t3              # get remainder into the t3
229 add t3, t3, 48        # convert the digit from decimal to binary
230 sb t3, 0(t0)          # store converted ascii digit into outputBuffer
231 sub t0, t0, 1         # update buffer address
232 bne t1, 0, while1    # loop back to while1 until t1 is not 0
233
234 addi t1, t0, 1        # copy a0 into a1
235
236 la t0, out1          # a0 = address of string out1
237 li t0, 4             # print out1 string
238 syscall
239
240 move t0, t1           # address of converted binary number to print
241 li t0, 4             # print t0 string
242 syscall
243
244 jx t0a              # return to main
245
246
247 # this function prints the input number as decimal
248 # t0 = input number as decimal
249 #printDec:
250 la t0, out2          # a0 = address of string out2
251 li t0, 4             # print out2 string
252 syscall
253
254 move t0, t0          # copy input number into a0
255 li t0, 36            # print number in a0 as unsigned decimal
256 syscall
257
258 jx t0a              # return to main
259
260 #this function converts decimal number to Hexadecimal and displays it in Hexadecimal format
261 # t0 = input number as decimal

```

Line: 150 Column: 1 Show Line Numbers

Mars Messages Run IO

Type here to search

36°C مشمس 9:04 PM 12/4/2023

Copro 0			Copro 1		
Name	Num	Value	Name	Num	Value
\$zero	0	0x00000000	\$t0	0	0x00000000
\$at	1	0x00000000	\$t1	1	0x00000000
\$v0	2	0x00000000	\$t2	2	0x00000000
\$t1	3	0x00000000	\$t3	3	0x00000000
\$a0	4	0x00000000	\$t4	4	0x00000000
\$a1	5	0x00000000	\$t5	5	0x00000000
\$a2	6	0x00000000	\$t6	6	0x00000000
\$a3	7	0x00000000	\$t7	7	0x00000000
\$t0	8	0x00000000	\$t8	8	0x00000000
\$t1	9	0x00000000	\$t9	9	0x00000000
\$t2	10	0x00000000	\$s0	10	0x00000000
\$t3	11	0x00000000	\$s1	11	0x00000000
\$t4	12	0x00000000	\$s2	12	0x00000000
\$t5	13	0x00000000	\$s3	13	0x00000000
\$t6	14	0x00000000	\$s4	14	0x00000000
\$t7	15	0x00000000	\$s5	15	0x00000000
\$t8	16	0x00000000	\$s6	16	0x00000000
\$t9	17	0x00000000	\$s7	17	0x00000000
\$s0	18	0x00000000	\$s8	18	0x00000000
\$s1	19	0x00000000	\$s9	19	0x00000000
\$s2	20	0x00000000	\$t0	20	0x00000000
\$s3	21	0x00000000	\$t1	21	0x00000000
\$s4	22	0x00000000	\$t2	22	0x00000000
\$s5	23	0x00000000	\$t3	23	0x00000000
\$s6	24	0x00000000	\$t4	24	0x00000000
\$s7	25	0x00000000	\$t5	25	0x00000000
\$s8	26	0x00000000	\$t6	26	0x00000000
\$s9	27	0x00000000	\$t7	27	0x00000000
\$t0	28	0x00000000	\$t8	28	0x00000000
\$t1	29	0x00000000	\$t9	29	0x00000000
\$t2	30	0x00000000	\$s0	30	0x00000000
\$t3	31	0x00000000	\$s1	31	0x00000000
\$t4	32	0x00000000	\$s2	32	0x00000000
\$t5	33	0x00000000	\$s3	33	0x00000000
\$t6	34	0x00000000	\$s4	34	0x00000000
\$t7	35	0x00000000	\$s5	35	0x00000000
\$t8	36	0x00000000	\$s6	36	0x00000000
\$t9	37	0x00000000	\$s7	37	0x00000000
\$s0	38	0x00000000	\$s8	38	0x00000000
\$s1	39	0x00000000	\$s9	39	0x00000000
\$s2	40	0x00000000	\$t0	40	0x00000000
\$s3	41	0x00000000	\$t1	41	0x00000000
\$s4	42	0x00000000	\$t2	42	0x00000000
\$s5	43	0x00000000	\$t3	43	0x00000000
\$s6	44	0x00000000	\$t4	44	0x00000000
\$s7	45	0x00000000	\$t5	45	0x00000000
\$s8	46	0x00000000	\$t6	46	0x00000000
\$s9	47	0x00000000	\$t7	47	0x00000000
\$t0	48	0x00000000	\$t8	48	0x00000000
\$t1	49	0x00000000	\$t9	49	0x00000000
\$t2	50	0x00000000	\$s0	50	0x00000000
\$t3	51	0x00000000	\$s1	51	0x00000000
\$t4	52	0x00000000	\$s2	52	0x00000000
\$t5	53	0x00000000	\$s3	53	0x00000000
\$t6	54	0x00000000	\$s4	54	0x00000000
\$t7	55	0x00000000	\$s5	55	0x00000000
\$t8	56	0x00000000	\$s6	56	0x00000000
\$t9	57	0x00000000	\$s7	57	0x00000000
\$s0	58	0x00000000	\$s8	58	0x00000000
\$s1	59	0x00000000	\$s9	59	0x00000000
\$s2	60	0x00000000	\$t0	60	0x00000000

File Edit Run Settings Tools Help

Run speed at max (no interaction)

```

1 # This function converts decimal number to Hexadecimal and displays it in Hexadecimal format
2 # t0 = input number as decimal
3 printHex
4 la t0, outputBuffer # 40 = address of output buffer
5 add t0, t0, 32 # 40 point to the end of outputBuffer
6 move t1, t0 # copy the decimal number in t1
7 li t2, 16 # t2 = 16
8 la t2, Hex # t2 = address of Hex
9
10 while:
11 divu t1, t2 # t1 / 16
12 mfu t1 # get quotient into t1
13 mfu t3 # get remainder into the t3
14 add t4, t3, t2 # convert the decimal to Hex digit
15 lb t5, 0(t4) # load the corresponding hex digit from Hex
16 sb t5, 0(t4) # store converted hex digit into outputBuffer
17 sub t4, t4, 1 # update buffer address
18 bne t1, 0, while # loop back to while until t1 is not 0
19
20 bne t3, '9', skip # check if HEX number will begin with letter
21 li t3, '0' # then expand a '0' before HEX number
22 sb t3, 0(t4)
23 move t4, t4 # copy 40 into a1
24 done
25
26 skip:
27 add t4, t4, 1 # copy 40 into a1
28
29 done:
30 la t0, out3 # 40 = address of string out3
31 li t0, 4 # print out3 string
32 syscall
33
34 move t4, t4 # address of converted hexadecimal number to print
35 li t0, 4 # print out3 string
36 syscall
37
38 jr t4 # return to main
39
40
41 # This function prints a newline as an output
42 Crlf:
43 la t0, newline # 40 = address of newline string
44 li t0, 4 # print a newline
45 syscall
46
47 jr t4 # return to main
48
49

```

Line: 150 Column: 1 Show Line Numbers

36°C مشمس 9:05 PM 12/4/2023

Output:

Mars Messages Run IO

This Program asks user to choose from menu a format for input number in a specified 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 4294967295) --> 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 0FFFFFFFh) --> A5
Binary = 10100101 Decimal = 165 Hexadecimal = A5

C3L|CCCS-217

Safanh saadi alzahrani 2210958

Danah saud alsubaie 2210611

```

Mars Messages Run IO
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 4294967296) --> 2392739283
Binary = 100011010011100100101011010011   Decimal = 2392739283   Hexadecimal = 8E84D3
1) Enter a binary Number
2) Enter a Decimal Number
3) Enter a Hexadecimal Number
4) Quit
Clear
Enter your choice --> 2
Enter a decimal number (less than 4294967296) --> 555555555
Error: Input too large. Try again please.
Enter your choice --> 5
Error: Invalid input.
Enter your choice --> 4
-- program is finished running --

```

eax	0x00000000
ecx	0x7fffffff
edx	0x00000000
ebx	0x00400000
esp	0x00400000
ebp	0x00000001
i386_	0x40030000

C3L|CCCS-217

Safanh saadi alzahrani 2210958

Danah saud alsubaie 2210611