

Lab Tasks

1. Convert by hand the number **-123456789** into its 32-bit single-precision binary representation, and Show your work for a full mark.

$$123\,456\,789 \div 2 = 61\,728\,394 + 1;$$

$$61\,728\,394 \div 2 = 30\,864\,197 + 0;$$

$$30\,864\,197 \div 2 = 15\,432\,098 + 1;$$

$$15\,432\,098 \div 2 = 7\,716\,049 + 0;$$

$$7\,716\,049 \div 2 = 3\,858\,024 + 1;$$

$$3\,858\,024 \div 2 = 1\,929\,012 + 0;$$

$$1\,929\,012 \div 2 = 964\,506 + 0;$$

$$964\,506 \div 2 = 482\,253 + 0;$$

$$482\,253 \div 2 = 241\,126 + 1;$$

$$241\,126 \div 2 = 120\,563 + 0;$$

$$120\,563 \div 2 = 60\,281 + 1;$$

$$60\,281 \div 2 = 30\,140 + 1;$$

$$30\,140 \div 2 = 15\,070 + 0;$$

$$15\,070 \div 2 = 7\,535 + 0;$$

$$7\,535 \div 2 = 3\,767 + 1;$$

$$3\,767 \div 2 = 1\,883 + 1;$$

$$1\,883 \div 2 = 941 + 1;$$

$$941 \div 2 = 470 + 1;$$

$$470 \div 2 = 235 + 0;$$

$$235 \div 2 = 117 + 1;$$

$$117 \div 2 = 58 + 1;$$

$$58 \div 2 = 29 + 0;$$

$$29 \div 2 = 14 + 1;$$

$$14 \div 2 = 7 + 0;$$

$$7 \div 2 = 3 + 1;$$

$$3 \div 2 = 1 + 1;$$

$$1 \div 2 = 0 + 1;$$

$$123\,456\,789 \text{ (decimal)} = 111\,0101\,1011\,1100\,1101\,0001\,0101 \text{ (binary)}$$

2. Convert by hand the floating-point number **1 10010100 100110000011000000000000** (shown in binary) into its corresponding decimal value. Show your work for a full mark.

First, we subtract 127 from the exponent to get the true exponent: $148 - 127 = 21$

Second, we convert the mantissa to decimal, by multiplying by 2^N of number:

$$1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 0 \times 2^{-6} + 0 \times 2^{-7} + 0 \times 2^{-8} + 0 \times 2^{-9} + 0 \times 2^{-10} +$$

$$1 \times$$

$$2^{-11} + 1 \times 2^{-12} + 0 \times 2^{-13} + 0 \times 2^{-14} + 0 \times 2^{-15} + 0 \times 2^{-16} + 0 \times 2^{-17} + 0 \times 2^{-18} + 0 \times 2^{-19} + 0 \times$$

$$2^{-20} + 0 \times 2^{-21} + 0 \times 2^{-22} + 0 \times 2^{-23} = 0.594482421875 \text{ (decimal)}$$

Then we will add 1 to the mantissa: $0.594482421875 \text{ (decimal)} + 1 = 1.594482421875$

(decimal)

Finally, the floating-point value is then equal to -1×2

$$21 \times 1.594482421875 = -3343872$$

3. Trace the following program by hand to determine the values of registers **\$f0** thru **\$f9**. Notice that **array1** and **array2** have the same elements, but in a different order. Comment on the sums of **array1** and **array2** elements computed in registers **\$f4** and **\$f9**, respectively. Now use the QTSPIM tool to trace the execution of the program and verify your results. What conclusion can be made from this exercise?

```
.data
array1: .float 5.6e+20, -5.6e+20, 1.2
array2: .float 1.2, 5.6e+20, -5.6e+20
.text
.globl main
```

```
main:
```

```
la    $t0, array1
lwc1  $f0, 0($t0)
lwc1  $f1, 4($t0)
lwc1  $f2, 8($t0)
add.s $f3, $f0, $f1
add.s $f4, $f2, $f3
la    $t1, array2
lwc1  $f5, 0($t1)
lwc1  $f6, 4($t1)
lwc1  $f7, 8($t1)
```

```
add.s  $f8, $f5, $f6
```

```
add.s  $f9, $f7, $f8
```

```
li $v0, 10      # To terminate the program
```

```
syscall
```

```
.end main
```

```
$f0 = 5.6e+20
```

```
$f1 = -5.6e+20
```

```
$f2 = 1.2
```

```
$f3 = $f0 + $f1 = 0
```

```
$f4 = $f2 + $f3 = 1.2
```

```
$f5 = 1.2
```

```
$f6 = 5.6e+20
```

```
$f7 = -5.6e+20
```

```
$f8 = $f5 + $f6 = 5.6e+20
```

```
$f9 = $f7 + $f8 = 0
```

The sum of elements of "array1" is 1.2 and the sum of elements of "array2" is 0. This shows that the order of elements in a floating-point addition operation can affect the result due to the limited precision of floating-point numbers.

4. Write an interactive program that inputs an integer **sum** and an integer **count**, computes, and displays the **average = (float) sum / (float) count** as a single-precision floating-point number. Hint: use the proper convert instruction to convert **sum** and **count** from integer word into single-precision float.

The image shows two screenshots of the MARS MIPS simulator. The first screenshot displays the initial assembly code for Lab8Task4.asm, which prompts the user for an integer sum and count, and begins the process of converting them to floats. The second screenshot shows the completed code, which calculates the average and prints the result.

First Screenshot: Initial Code

```

1 .data
2 sum: .asciiz "Enter sum: "
3 count: .asciiz "Enter count: "
4 result: .asciiz "Average: "
5 .text
6 # Print prompt for sum
7 li $v0, 4
8 la $a0, sum
9 syscall
10 # Read sum
11 li $v0, 5
12 syscall
13 move $s0, $v0
14 # Print prompt for count
15 li $v0, 4
16 la $a0, count
17 syscall
18 # Read count
19 li $v0, 5
20 syscall
21 move $s1, $v0
22 # Convert integers to float
23 mtcl $s0, $f0
24 mtcl $s1, $f2
25 cvt.s.w $f0, $f0

```

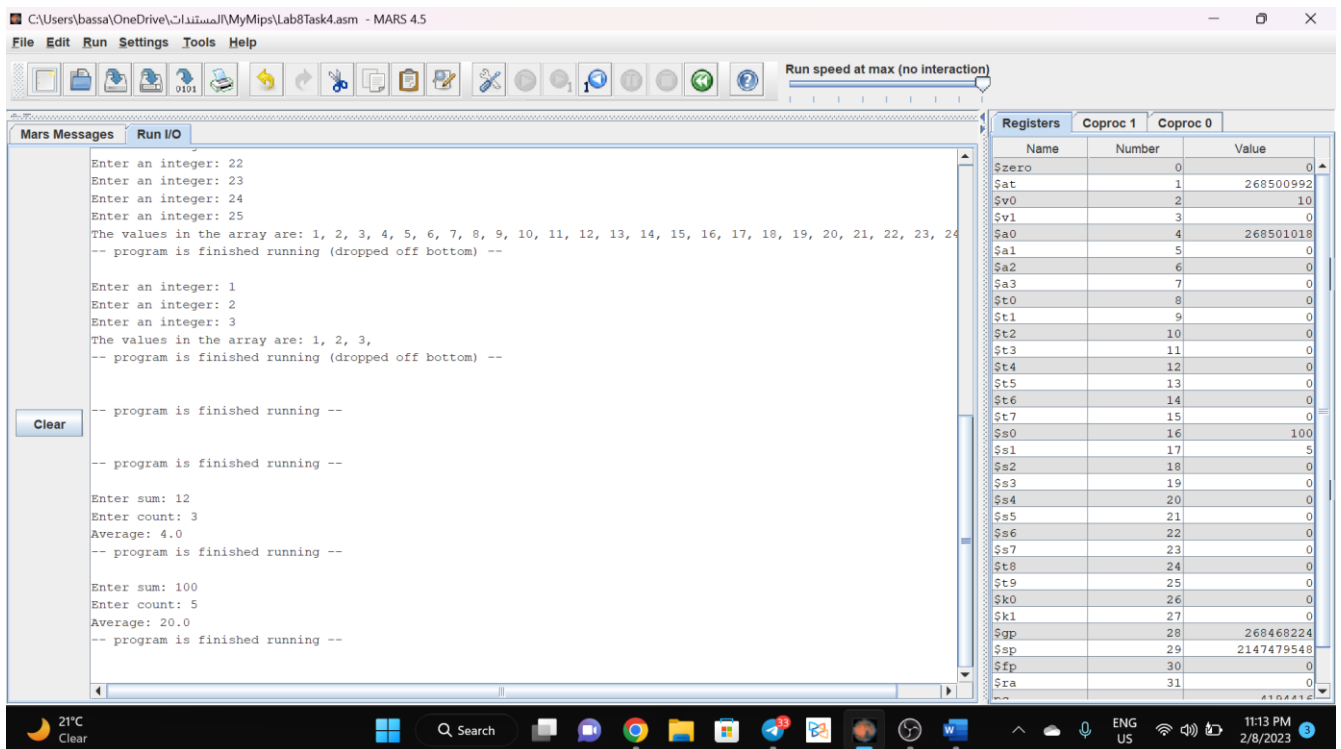
Second Screenshot: Completed Code

```

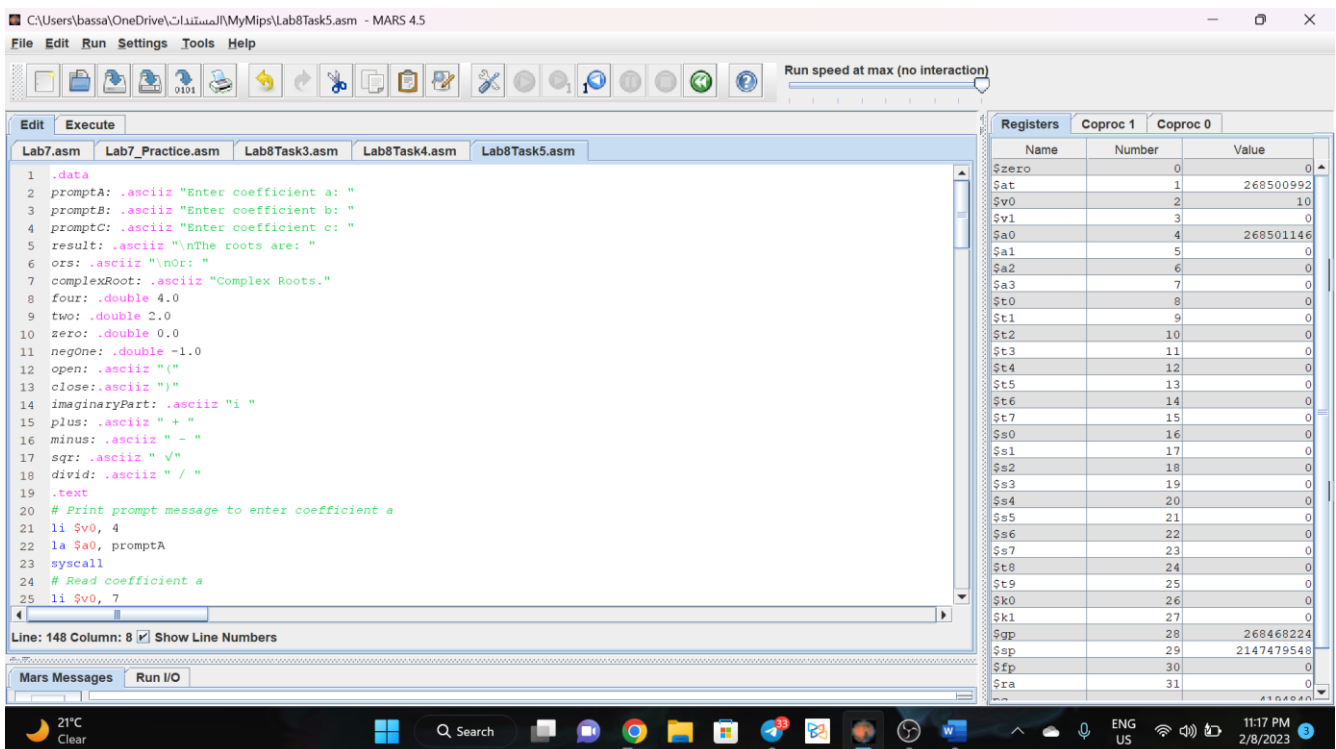
14 # Print prompt for count
15 li $v0, 4
16 la $a0, count
17 syscall
18 # Read count
19 li $v0, 5
20 syscall
21 move $s1, $v0
22 # Convert integers to float
23 mtcl $s0, $f0
24 mtcl $s1, $f2
25 cvt.s.w $f0, $f0
26 cvt.s.w $f2, $f2
27 # Compute average
28 div.s $f4, $f0, $f2
29 # Print result
30 li $v0, 4
31 la $a0, result
32 syscall
33 li $v0, 2
34 mov.s $f12, $f4
35 syscall
36 li $v0, 10
37 syscall

```

The registers window on the right of each screenshot shows the state of the MIPS registers. In the first screenshot, \$s0 and \$s1 are 0. In the second screenshot, \$s0 is 268500992 and \$s1 is 268501018, which are the integer values of sum and count respectively. The floating-point registers \$f0 and \$f2 contain the converted values.



- Write an interactive program that inputs the coefficient of a quadratic equation, computes, and displays the roots of the quadratic equation. All input, computation, and output should be done using double-precision floating-point instructions and registers. The program should handle the case of complex roots and displays the results properly.



C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm

```

25 li $v0, 7
26 syscall
27 mov.d $f20, $f0
28 # Print prompt message to enter coefficient b
29 li $v0, 4
30 la $a0, promptB
31 syscall
32 # Read coefficient b
33 li $v0, 7
34 syscall
35 mov.d $f14, $f0
36 # Print prompt message to enter coefficient c
37 li $v0, 4
38 la $a0, promptC
39 syscall
40 # Read coefficient c
41 li $v0, 7
42 syscall
43 mov.d $f16, $f0
44 # Calculate the discriminant: $f8 = $f14^2 - 4 * $f12 * $f16
45 mul.d $f8, $f14, $f14 # $f8 = $f14 * $f14
46 ldc1 $f10, four # load four into $f10
47 mul.d $f10, $f10, $f20 # 4 * $f20
48 mul.d $f10, $f10, $f16 # $f10 = $f10 * $f16
49 sub.d $f8, $f8, $f10 # $f8 = $f8 - $f10

```

Line: 22 Column: 16 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501146
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

21°C Clear

Search

ENG US

11:17 PM 2/8/2023

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm

```

49 sub.d $f8, $f8, $f10 # $f8 = $f8 - $f10
50 # Check if the roots are complex
51 ldc1 $f18, zero
52 c.lt.d $f8, $f18
53 bclt complex
54 # Compute the roots
55 sqrt.d $f8, $f8 # square root $f8 and store it in $f8
56 ldc1 $f10, two
57 mul.d $f6, $f10, $f20
58 sub.d $f14, $f18, $f14
59 add.d $f4, $f14, $f8
60 div.d $f22, $f4, $f6 # $f10 = $f8 / $f6
61
62
63 # Print the roots
64 li $v0, 4
65 la $a0, result
66 syscall
67 li $v0, 3
68 mov.d $f12, $f22
69 syscall
70 li $v0, 4
71 la $a0, ors
72 syscall
73 li $v0, 3

```

Line: 40 Column: 21 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501146
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

21°C Clear

Search

ENG US

11:18 PM 2/8/2023

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm

```

73 li $v0, 3
74 sub.d $f4, $f14, $f8
75 div.d $f12, $f4, $f6
76 syscall
77 j end
78 # Handle complex roots
79 complex:
80 ldcl $f6 negOne
81 mul.d $f4, $f8 $f6
82 sqrt.d $f6, $f4
83 sub.d $f14, $f18, $f14 # converting b to -b
84 ldcl $f12, two
85 mul.d $f8, $f12, $f20
86
87 # Print the roots (part real, part imaginary)
88 li $v0, 4
89 la $a0, result
90 syscall
91 li $v0, 4
92 la $a0, open
93 syscall
94 li $v0, 3
95 mov.d $f12, $f14
96 syscall
97 li $v0, 4

```

Line: 40 Column: 21 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501146
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

21°C Clear

11:18 PM 2/8/2023

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm

```

97 li $v0, 4
98 la $a0, plus
99 syscall
100 li $v0, 3
101 mov.d $f12, $f6
102 syscall
103 li $v0, 4
104 la $a0, imaginaryPart
105 syscall
106 li $v0, 4
107 la $a0, close
108 syscall
109 li $v0, 4
110 la $a0, divid
111 syscall
112 li $v0, 3
113 mov.d $f12, $f8
114 syscall
115 # printing the minus case
116 li $v0, 4
117 la $a0, ors
118 syscall
119 li $v0, 4
120 la $a0, result
121 syscall

```

Line: 40 Column: 21 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501146
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

21°C Clear

11:18 PM 2/8/2023

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task5.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm

```
125 li $v0, 3
126 mov.d $f12, $f14
127 syscall
128 li $v0, 4
129 la $a0, minus
130 syscall
131 li $v0, 3
132 mov.d $f12, $f6
133 syscall
134 li $v0, 4
135 la $a0, imaginaryPart
136 syscall
137 li $v0, 4
138 la $a0, close
139 syscall
140 li $v0, 4
141 la $a0, divid
142 syscall
143 li $v0, 3
144 mov.d $f12, $f8
145 syscall
146 end:
147 li $v0, 10
148 syscall
149
```

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501146
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

Line: 40 Column: 21 ☒ Show Line Numbers

Mars Messages Run I/O

21°C Clear

Search

ENG US

11:19 PM 2/8/2023

6. Square Root Calculation: Newton's iterative method can be used to approximate the square root of a number **x**. Let the initial **guess** be **1**. Then each new **guess** can be computed as follows:

$$\text{guess} = ((x/\text{guess}) + \text{guess}) / 2;$$

Write a function called **square_root** that receives a double-precision parameter **x**, computes, and returns the approximated value of the square root of **x**. Write a loop that repeats 20 times and computes 20 **guess** values, then returns the final **guess** after 20 iterations. Use the MIPS floating-point register convention to pass the parameter **x** and to return the function result. All computation should be done using double-precision floating-point instructions and registers. Compare the result of the **sqrtd** instruction against the result of your **square_root** function. What is the error in absolute value?

The screenshot shows the MARS MIPS simulator interface. The main window displays the assembly code for Lab8Task6.asm. The code includes data declarations for messages, a number, a guess, and a loop counter. The main function prints the number and guess, then calls the square_root function. The register window on the right shows the state of the MIPS registers, with the \$f12 register containing the value 2.68500992.

```

1 .data
2 sqnum: .asciiz "The number we are going to take the square root is: "
3 guessnum: .asciiz "\nThe guess number is: "
4 sqz: .asciiz "\nThe sqrt.d answer is: "
5 num: .double 5.43987564387543E14
6 guess: .double 1.0
7 half: .double 0.5
8 twenty: .word 20
9 .text
10 main:
11 # Print message
12 li $v0, 4
13 la $a0, sqnum
14 syscall
15 # Print number
16 li $v0, 3
17 l.d $f12, num
18 syscall
19 # Print guess message
20 li $v0, 4
21 la $a0, guessnum
22 syscall
23 # Call square_root function
24 j square_root
25 square_root:

```

Register	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501068
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task6.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm Lab8Task6.asm

```
26 # Load initial guess
27 l.d $f0, guess
28 # Load loop counter
29 lw $t0, twenty
30 # Repeat 20 times
31 loop:
32 # Calculate new guess
33 l.d $f4, num
34 div.d $f6, $f4, $f0
35 add.d $f6, $f6, $f0
36 l.d $f2, half
37 mul.d $f6, $f6, $f2
38 # Store new guess in $f0
39 mov.d $f0, $f6
40 # Decrement loop counter
41 addi $t0, $t0, -1
42 bne $t0, $zero, loop
43 # Print result
44 li $v0, 3
45 mov.d $f12, $f0
46 syscall
47 # Compare with sqrt.d
48 li $v0, 4
49 la $a0, sqt
50 syscall
```

Line: 56 Column: 8 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	268501068
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

21°C Clear

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task6.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm Lab8Task6.asm

```
32 # Calculate new guess
33 l.d $f4, num
34 div.d $f6, $f4, $f0
35 add.d $f6, $f6, $f0
36 l.d $f2, half
37 mul.d $f6, $f6, $f2
38 # Store new guess in $f0
39 mov.d $f0, $f6
40 # Decrement loop counter
41 addi $t0, $t0, -1
42 bne $t0, $zero, loop
43 # Print result
44 li $v0, 3
45 mov.d $f12, $f0
46 syscall
47 # Compare with sqrt.d
48 li $v0, 4
49 la $a0, sqt
50 syscall
51 li $v0, 3
52 sqrt.d $f12, $f4
53 syscall
54 # Exit program
55 li $v0, 10
56 syscall
```

Line: 56 Column: 8 ☒ Show Line Numbers

Mars Messages Run I/O

Registers	Coproc 1	Coproc 0
Name	Float	Double
\$f0	-0.42282248	5.1913644684564495E8
\$f1	23.867867	
\$f2	0.0	0.5
\$f3	1.75	
\$f4	-3.975701E-24	5.43987564387543E14
\$f5	127.46105	
\$f6	-0.42282248	5.1913644684564495E8
\$f7	23.867867	
\$f8	0.0	0.0
\$f9	0.0	
\$f10	0.0	0.0
\$f11	0.0	
\$f12	7.4375025E9	2.3323540991614953E7
\$f13	15.390191	
\$f14	0.0	0.0
\$f15	0.0	
\$f16	0.0	0.0
\$f17	0.0	
\$f18	0.0	0.0
\$f19	0.0	
\$f20	0.0	0.0
\$f21	0.0	
\$f22	0.0	0.0
\$f23	0.0	
\$f24	0.0	0.0
\$f25	0.0	
\$f26	0.0	0.0
\$f27	0.0	

Condition Flags

☐ 0 ☐ 1 ☐ 2 ☐ 3

☐ 4 ☐ 5 ☐ 6 ☐ 7

21°C Clear

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task6.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Lab7.asm Lab7_Practice.asm Lab8Task3.asm Lab8Task4.asm Lab8Task5.asm Lab8Task6.asm

C:\Users\bassa\OneDrive\المستندات\MyMips\Lab8Task6.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Mars Messages Run I/O

```

The number we are going to take the square root is: 5.43987564387543E14
The guess number is: 5.1913644684564495E8
The sqrt.d answer is: 2.3323540991614953E7
-- program is finished running --

```

Clear

Registers		Coproc 1	Coproc 0
Name	Float	Double	
\$f0	-0.42282248	5.1913644684564495E8	
\$f1	23.867867		
\$f2	0.0	0.5	
\$f3	1.75		
\$f4	-3.975701E-24	5.43987564387543E14	
\$f5	127.46105		
\$f6	-0.42282248	5.1913644684564495E8	
\$f7	23.867867		
\$f8	0.0	0.0	
\$f9	0.0		
\$f10	0.0	0.0	
\$f11	0.0		
\$f12	7.4375025E9	2.3323540991614953E7	
\$f13	15.390191		
\$f14	0.0	0.0	
\$f15	0.0		
\$f16	0.0	0.0	
\$f17	0.0		
\$f18	0.0	0.0	
\$f19	0.0		
\$f20	0.0	0.0	
\$f21	0.0		
\$f22	0.0	0.0	
\$f23	0.0		
\$f24	0.0	0.0	
\$f25	0.0		
\$f26	0.0	0.0	
\$f27	0.0		
\$f28	0.0	0.0	

Condition Flags

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

21°C Clear

Search

ENG US

11:23 PM 2/8/2023

The difference gets bigger when the number is big. We can fix that by increasing the loops from 20 to something bigger.