

EXERCISE**1. Addition of Two Integers**

Write a MIPS program to add **15** and **25**, then print the result.

```
.data
msg: .asciiz "The result of the operation is: "

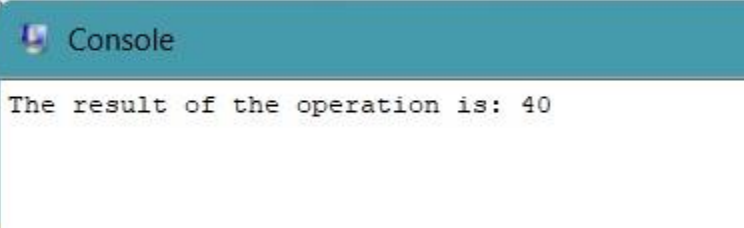
.text
.globl main

main:
    li $t0, 15
    li $t1, 25
    add $t2, $t0, $t1

    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 1
    move $a0, $t2
    syscall

    li $v0, 10
    syscall
```

Output

```
r data segment [10000000]..[10040000]
000000
010000
010010
010020 The result of the operation is: 40
n Stop
```

2. Subtraction of Two Integers

Write a MIPS program to subtract **50** from **75** and print the result.

```
.data
msg: .asciiz "The result of the operation is: "

.text
.globl main

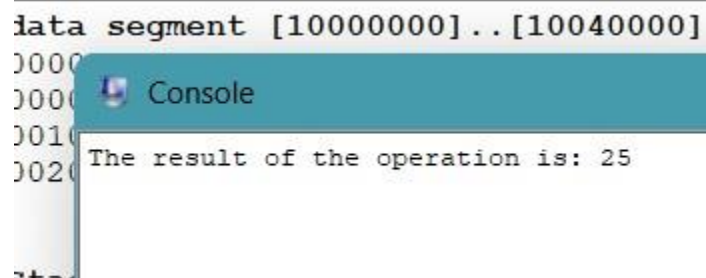
main:
    li $t0, 75
    li $t1, 50
    sub $t2, $t0, $t1

    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 1
    move $a0, $t2
    syscall

    li $v0, 10
    syscall
```

Output



```
data segment [10000000]..[10040000]
0000
0000
0010
0020 The result of the operation is: 25
...
```

3. Complex Equation (Addition and Multiplication)

Write a MIPS program to calculate and print the result of:
 $(10+20) \times 5$

```
.data
msg: .asciiz "The result of the operation is: "

.text
.globl main

main:
    li $t0, 10
    li $t1, 20
    add $t2, $t0, $t1    # t2 = 10 + 20

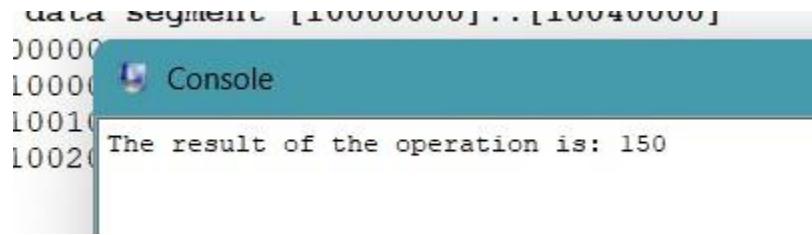
    li $t3, 5
    mul $t4, $t2, $t3    # t4 = (10 + 20) * 5

    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 1
    move $a0, $t4
    syscall

    li $v0, 10
    syscall
```

Output



```
data segment [10000000]..[10040000]
00000
10000
10010
10020 The result of the operation is: 150
```

4. Complex Equation (Subtraction and Division)

Write a MIPS program to calculate and print the result of:

$(50-30)/2$ the result of the operation is:" followed by an integer result, e.g., 42.

```
.data
msg: .asciiz "The result of the operation is: "

.text
.globl main

main:
    li $t0, 50
    li $t1, 30
    sub $t2, $t0, $t1    # t2 = 50 - 30

    li $t3, 2
    div $t2, $t3
    mflo $t4             # result in t4

    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 1
    move $a0, $t4
    syscall

    li $v0, 10
    syscall
```

```
c data segment [10000000]..[10040000]
000000
010000
010010
010020 The result of the operation is: 10

c Stack
ffff00
```

5. String with Float

Write a MIPS program to print the message:

"The floating-point result is:" followed by a floating-point value, e.g., 3.14.

```
.data
msg: .asciiz "The floating-point result is: "
val: .float 3.14

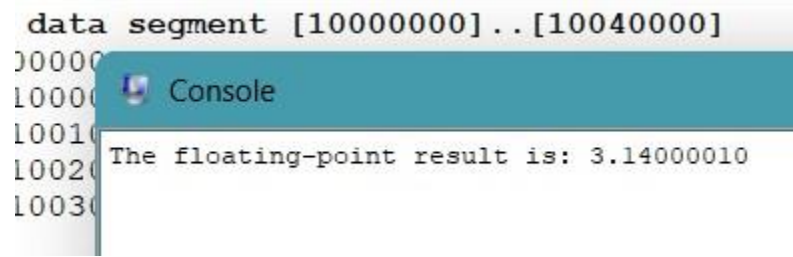
.text
.globl main

main:
    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 2
    l.s $f12, val
    syscall

    li $v0, 10
    syscall
```

Output



The screenshot shows a MIPS simulator interface. On the left, a list of memory addresses is displayed: 10000, 10000, 10010, 10020, and 10030. A teal-colored console window is overlaid on the right, displaying the output of the program: "The floating-point result is: 3.14000010".

6. Equation with Multiple Operations

Write a MIPS program to compute and print the result of the equation:

$$((10+5) \times 4) - 15$$

```
.data
msg: .asciiz "The result of the operation is: "

.text
.globl main

main:
    li $t0, 10
    li $t1, 5
    add $t2, $t0, $t1    # t2 = 10 + 5

    li $t3, 4
    mul $t4, $t2, $t3    # t4 = (10 + 5) * 4

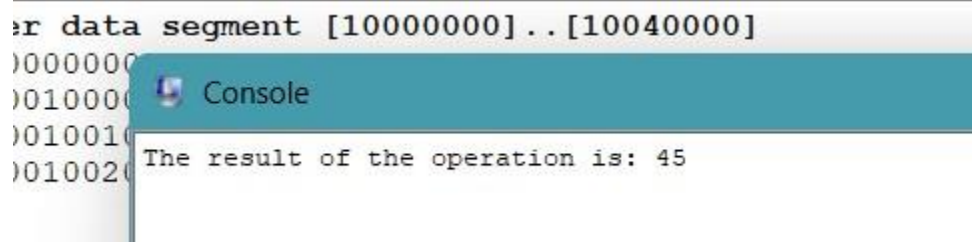
    li $t5, 15
    sub $t6, $t4, $t5    # t6 = ((10 + 5) * 4) - 15

    li $v0, 4
    la $a0, msg
    syscall

    li $v0, 1
    move $a0, $t6
    syscall

    li $v0, 10
    syscall
```

Output



The screenshot shows a MIPS simulator interface. On the left, a list of memory addresses is displayed: 00000000, 00100000, 00100100, and 00100200. A console window titled "Console" is open, displaying the output of the program: "The result of the operation is: 45".