

# MIPS Assembly Calculator User Manual

## Table of Contents

Introduction (2)

Operating the Calculator (2)

Addition (3)

Subtraction (4)

Multiplication (5)

Division (7)

Modulo (8)

Square (9)

Exponentiation (10)

Factorial (12)

Exiting the Application (13)

## Introduction

This application written in MIPS assembly language is a 32-bit calculator that performs 8 common mathematical operations with integers. The application can be run in a SPIM simulator; the developer used QtSpim for testing and operation of the calculator.

## Operating the Calculator

Upon opening the source file in the SPIM simulator the main menu of the calculator is displayed in the console and prompts the user to select the desired operation. The menu appears as below:

```
Welcome to the 32 bit integer Calculator!
Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
|
```

*Fig. 1: calculator menu as it appears upon launch*

A welcome message appears, and the list of possible operations is displayed. At this point the user is prompted to type the associated menu number of the desired operation. For example, to perform addition the user inputs "1" and presses the ENTER key. The user can select any operation numbered 1-8 or exit the application with input "0" and ENTER. Detailed information about each individual operation is provided in this manual.

All input prompts are handled the same way. The calculator is designed to take integer input only. The user interacts with the menu entirely through integer input, and all operations are performed using integer input. Simply typing the desired input into the console and pressing ENTER is all that is needed to operate this program.

Operands are referred to as "x" and "y" for clarity within the calculator interface. Operations that take 2 operands (addition, subtraction, multiplication, division, modulo, and exponential) prompt the user to enter both an x and y value. Operations that take only 1 operand (square and factorial) only prompt the user to enter a single operand x.

### Addition

This function is accessed when the user enters “1” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
1
The chosen operation is x + y
Enter a value for x:
|

```

*Fig. 2: program prompts user to input the first operand value for addition*

The program then prompts the user to enter their second operand.

```

1
The chosen operation is x + y
Enter a value for x:
5
Enter a value for y:
5
5 + 5 = 10

```

*Fig. 3: calculator display after completing the addition operation*

As seen in Fig. 3 above, when the calculator receives the second operand input from the user the result of the addition is automatically displayed below. The calculator will display confirmation of the user’s operands and the result of the addition. After the result is displayed, the program will automatically return to the main menu.

Positive and negative integer operands are allowed in the addition function.

### Subtraction

This function is accessed when the user enters “2” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
2
The chosen operation is x - y
Enter a value for x:
|

```

*Fig. 4: program prompts user to input the first operand value for subtraction*

The program then prompts the user to enter the value of the second operand.

```

2
The chosen operation is x - y
Enter a value for x:
5
Enter a value for y:
7
5 - 7 = -2

```

*Fig. 5: calculator display after completing the subtraction operation*

As seen in Fig. 5 above, when the calculator receives the second operand input from the user the result of the subtraction is automatically displayed below. The calculator will display confirmation of the user’s operands and the result of the subtraction. After the result is displayed, the program will automatically return to the main menu.

Note as in the example shown in Fig. 5 the order of operands does matter, the calculator will perform  $x-y$  as entered by the user. Subtraction allows both positive and negative integer operands.

### Multiplication

This function is accessed when the user enters “3” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
3
The chosen operation is x * y
Enter a value for x:
|

```

*Fig. 6: program prompts user to input the first operand value for multiplication*

The program then prompts the user to enter the value of the second operand.

```

3
The chosen operation is x * y
Enter a value for x:
5
Enter a value for y:
-1
5 * -1 = -5

```

*Fig. 7: calculator display after completing the multiplication operation*

As seen in Fig. 7 above, when the calculator receives the second operand input from the user the result of the multiplication is automatically displayed below. The calculator will display confirmation of the user’s operands and the result of the multiplication. After the result is displayed, the program will automatically return to the main menu.

Positive and negative integer operands are allowed as seen in the example in Fig. 7.

## Division

Please note that this program only performs integer division, meaning the result of the calculation will be truncated to integers. Because of this, functionality has been added to show when the calculation results in a non-zero remainder. Calculations that do and do not result in non-zero remainders are shown below.

This function is accessed when the user enters "4" on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```
Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
4
The chosen operation is x / y
Enter a value for x:
|
```

*Fig. 8: program prompts user to input the first operand (dividend) value for division*

The program then prompts the user to enter the value of the second operand (divisor).

```
4
The chosen operation is x / y
Enter a value for x:
8
Enter a value for y:
2
8 / 2 = 4
```

*Fig. 9: calculator display after completing the division operation without a remainder*

As seen in Fig. 9 above, when the calculator receives the second operand input from the user the result of the division is automatically displayed below. The calculator will display confirmation of the user's operands and the result of the division. After the result is displayed, the program will automatically return to the main menu.

Positive and negative operands are allowed in the division function.

See Fig. 10 below for a second example output showing the results of a calculation that produces a remainder.

```
4
The chosen operation is x / y
Enter a value for x:
9
Enter a value for y:
2
9 / 2 = 4 with a remainder of 1
```

*Fig. 10: calculator display after completing the division operation with a non-zero remainder*

The interface is exactly the same as before except for the remainder message that has been appended to the end of the result. This feature ensures that the user is always aware when the division operation produces a non-zero remainder. As in the example above, after displaying the calculation result the program will automatically return to the main menu.

## Modulo

This function is accessed when the user enters “5” on the main menu. Modulo calculates the remainder of a division operation. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
5
The chosen operation is x % y
Enter a value for x:
|

```

*Fig. 11: program prompts user to input the first operand (dividend) value for modulo*

The program then prompts the user to enter the value of the second operand (divisor).

```

5
The chosen operation is x % y
Enter a value for x:
9
Enter a value for y:
2
9 % 2 = 1

```

*Fig. 12: calculator display after completing the modulo operation*

As seen in Fig. 12 above, when the calculator receives the second operand input from the user the result of the modulo is automatically displayed below. The calculator will display confirmation of the user’s operands and the result of the modulo. After the result is displayed, the program will automatically return to the main menu.

Positive and negative operands are allowed in the modulo calculation. User may note the relationship between the modulo operation and division, the results in Fig. 10 and Fig. 12 illustrate this relationship. The division operation will display the remainder only when it is non-0. Because of this, users should choose the modulo operation when seeking the remainder specifically.



### Square

This function is accessed when the user enters “6” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first and only operand.

```
Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
6
The chosen operation is x ^ 2
Enter a value for x:
5
5 ^ 2 = 25
```

*Fig. 13: calculator display after completing the square operation*

Because the square operation only takes a single operand input from the user, the result is displayed immediately after receiving the initial user input. The calculator will display confirmation of the user's operand and its square. After the result is displayed, the program will automatically return to the main menu.

Positive and negative operands are allowed in the square calculation. Note that square will always return a positive result as any integer multiplied by itself will result in a positive value.

### Exponentiation

This function is accessed when the user enters “7” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
7
The chosen operation is x ^ y
Enter a value for x:
|

```

*Fig. 14: program prompts user to input the first operand (base) value for exponentiation*

The program then prompts the user to enter the value of the second operand (exponent).

```

7
The chosen operation is x ^ y
Enter a value for x:
5
Enter a value for y:
3
5 ^ 3 = 125

```

*Fig. 15: calculator display after completing the exponentiation operation*

As seen in Fig. 15 above, when the calculator receives the second operand input from the user the result of the exponentiation is automatically displayed below. The calculator will display confirmation of the user’s operands and the result of the exponentiation. After the result is displayed, the program will automatically return to the main menu.

Positive and negative operands are allowed for the first operand, the base. However, negative exponents are not allowed for the exponent operand. When the user enters an invalid value for the exponent, the calculator displays an error message as seen in Fig. 16:

```
7
The chosen operation is x ^ y
Enter a value for x:
5
Enter a value for y:
-2
Negative exponents are not allowed, please try again.
The chosen operation is x ^ y
Enter a value for x:
|
```

---

*Fig. 16: error message upon receiving a negative exponent value*

The message “Negative exponents are not allowed, please try again.” is displayed, and the calculator returns to the beginning of the exponentiation operation. From there the user must enter their chosen base again and then input a non-negative exponent.

### Factorial

This function is accessed when the user enters “8” on the main menu. The calculator will return a message confirming what the selected operation is, and then ask the user to enter the first and only operand.

```

Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
8
The chosen operation is x!
Enter a value for x:
5
5! = 120

```

*Fig. 17: calculator display after completing the factorial operation*

Because the factorial operation only takes a single operand input from the user, the result is displayed immediately after receiving the user input. The calculator will display confirmation of the user's operand and the resulting value. After the result is displayed, the program will automatically return to the main menu.

Negative operands are not allowed for factorial operations, as  $x!$  is undefined for  $x < 0$ . Upon receiving an invalid negative input for the factorial operand, the program will display the following error message:

```

8
The chosen operation is x!
Enter a value for x:
-4
Negative factorial is undefined, please enter a positive integer.
The chosen operation is x!
Enter a value for x:
4
4! = 24

```

*Fig. 18: error message indicating that negative operands are not allowed for factorial*

As seen in Fig. 18, after the error message is displayed the calculator once again prompts the user to enter a valid integer value.

### Exiting the Application

Closing the application can be done anytime the user is interfacing with the main menu. The user enters "0", and the program displays the exit message and terminates.

```
Enter the number of the operation to be performed
(1) x + y
(2) x - y
(3) x * y
(4) x / y
(5) x % y
(6) x ^ 2
(7) x ^ y
(8) x!
(0) EXIT
0
Exiting the Calculator. Thank you!
|
```

*Fig. 19: exit message displayed upon closing the program.*

The program can only be closed from the main menu. If the user wishes to close the program, they must complete the current calculation upon which they will be returned to the main menu automatically.