

First Program

Step 1 : Save the following code in a Notepad file with ‘.s’ extension.

Program to add two numbers.

```
.data      #.data section is required so as to initialize ordinary data, in this case it is a message
MSG1: .asciiz " The numbers 5 and 6 are added together : " # MSG1 is a label then it is followed by
data type .asciiz, followed by message to
be printed on to the console window

.text      # It is the place where you start writing source code in the program

main:      #Every execution of the MIPS program starts with main label followed by colon

    li $t1,5      # loads the immediate value 5 into $t1,t1 is a temporary register and $t1 indicates
the content of that register]

    li $t2,6      # loads the immediate value 6 into $t2,t2 is another temporary register and $t2
indicates the content of that register]

    add $t1,$t1,$t2  # $t1=$t1+$t2 # Adding the contents of the register t1 and t2 and finally
putting the content to destination register t1 itself

    li $v0,4      # Print out a string , Special register v0 to print anything to the console. If value 4
is passed, we are printing a string to the console

    la $a0,MSG1    # Instruction is load address, that means it is now pointing to the MSG1.ie,Get
address of a string MSG1 into a0 as content. a0 is another special register used for this purpose.

    syscall      # Printing the string from a0 to the console

    li $v0,1      # Print out a integer ,Special register v0 to print anything to the console. If value 1
is passed, we are printing a integer to the console

    move $a0,$t1    # Move the number to a0 from the destination register t1, since t1 now
contains the result of addition

    syscall      # display the integer

    jr $ra      # ra is a sprcial register which holds the return address. Setting Program Counter
as $ra, Return Address Content
```

Program in Notepad

```
add - Notepad
File Edit Format View Help

# Program to add two numbers.

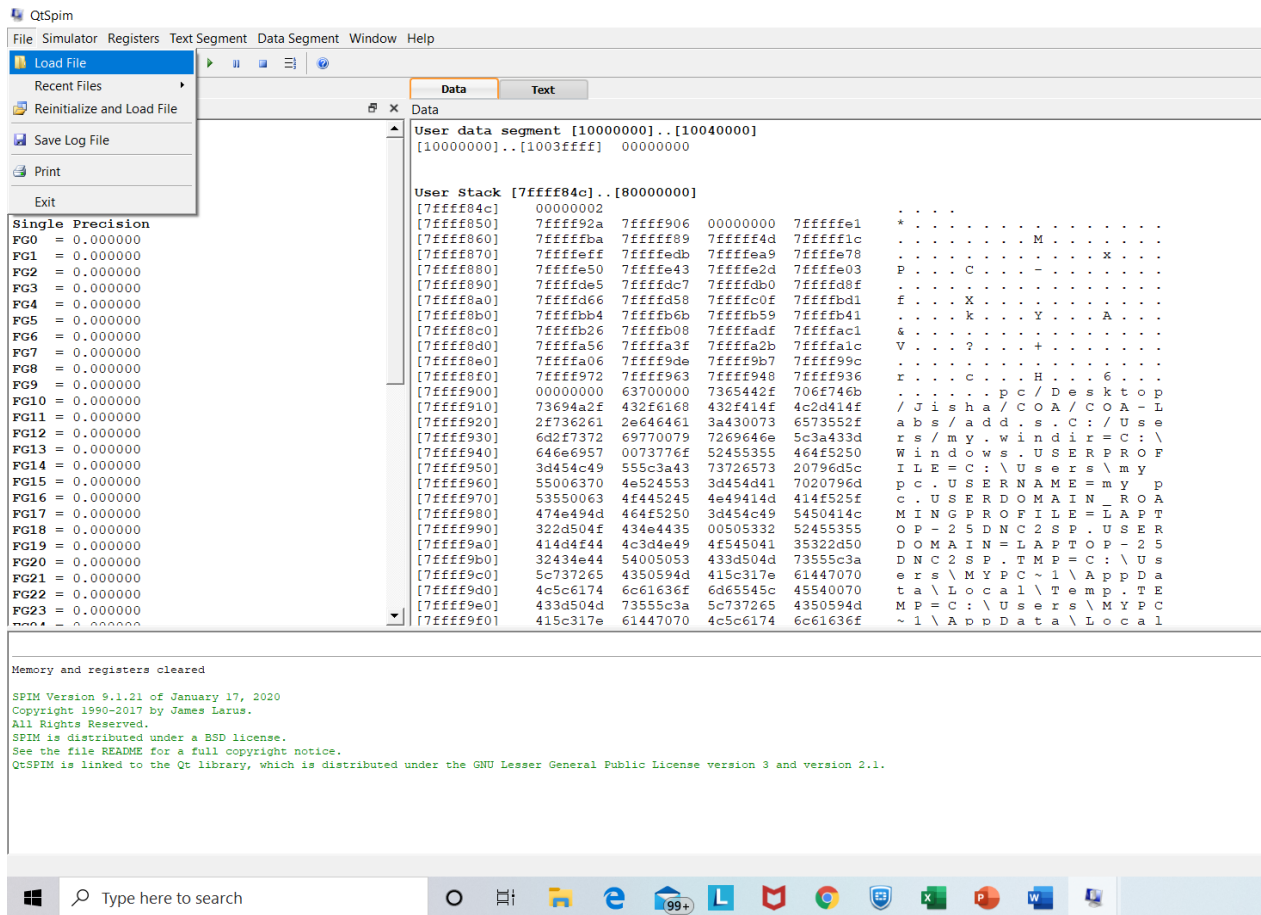
.data
MSG1: .asciiz " The numbers 5 and 6 are added together : "
.text

main:
    li $t1,5           # loads the value 5 into $t1
    li $t2,6           # loads the value 6 into $t2
    add $t1,$t1,$t2     # $t1=$t1+$t2
    li $v0,4           # Print out a string
    la $a0,MSG1         # Get address of a string
    syscall            # Printing the string
    li $v0,1           # Tell syscall to print a number
    move $a0,$t1        # Move the number to $a0
    syscall            # display it
    jr $ra             # ra holds the return address. Setting PC as $ra
```

Ln 18, Col 47 100% Windows (CRLF) UTF-8

Type here to search 99+ 12:07 17-08-2020 ENG 22

Step 2 : Open QtSpim simulator and load the file .



Step 3 : Run the program . See the screenshot .

QtSpim

File Simulator Registers Text Segment Data Segment Window Help

FP Regs Int Regs [10] Run/Continue Data Text

FP Regs

FIR = 38912
FCSR = 0
FCCR = 0
FEXR = 0

Single Precision
FG0 = 0.000000
FG1 = 0.000000
FG2 = 0.000000
FG3 = 0.000000
FG4 = 0.000000
FG5 = 0.000000
FG6 = 0.000000
FG7 = 0.000000
FG8 = 0.000000
FG9 = 0.000000
FG10 = 0.000000
FG11 = 0.000000
FG12 = 0.000000
FG13 = 0.000000
FG14 = 0.000000
FG15 = 0.000000
FG16 = 0.000000
FG17 = 0.000000
FG18 = 0.000000
FG19 = 0.000000
FG20 = 0.000000
FG21 = 0.000000
FG22 = 0.000000
FG23 = 0.000000

User data segment [10000000]..[10040000]

```
[10000000]..[1000ffff] 00000000
[10010000] 65685420 6d756e20 73726562 61203520 The numbers 5 a
[10010010] 3620646e 65726120 64646120 74206465 nd 6 are added t
[10010020] 7465676f 20726568 0000203a 00000000 ogether : .....
[10010030]..[1003ffff] 00000000
```

User Stack [7ffff84c]..[80000000]

```
[7ffff84c] 00000002 .....
[7ffff850] 7ffff92a 7ffff906 00000000 7fffffe1 * .....
[7ffff860] 7ffff8ba 7ffff8f9 7fffff4d 7fffff1c .....M .....
[7ffff870] 7ffffeff 7ffffedb 7ffffea9 7ffffe78 .....x .....
[7ffff880] 7ffffe50 7ffffe43 7ffffe2d 7ffffe03 P...C...- .....
[7ffff890] 7ffffde5 7ffffdc7 7ffffdb0 7ffffd8f .....
[7ffff8a0] 7ffffd66 7ffffd58 7ffffc0f 7ffffbd1 f...X .....
[7ffff8b0] 7ffffb4 7ffffb6b 7ffffb59 7ffffb41 .....k...Y...A...
[7ffff8c0] 7ffffb26 7ffffb08 7ffffadf 7ffffac1 & .....
[7ffff8d0] 7ffffa56 7ffffa3f 7ffffa2b 7ffffa1c V...?...+ .....
[7ffff8e0] 7ffffa06 7ffff9de 7ffff9b7 7ffff99c .....
[7ffff8f0] 7ffff972 7ffff963 7ffff948 7ffff936 r...c...H...6...
[7ffff900] 00000000 63700000 7365442f 706f746b .....pc/Desktop
[7ffff910] 73694a2f 432f6168 432f414f 4c2d414f /Jisha/COA/COA-L
[7ffff920] 2f736261 2e646461 3a430073 6573552f abs/add.s.C:/Use
[7ffff930] 6d2f7372 69770079 7269646e 5c3a433d rs/my.windir=C:\
[7ffff940] 646e6957 0073776f 52455355 464f5250 Windows.USERPROF
[7ffff950] 3d454c49 555c3a43 73726573 20796d5c ILE=C:\Users\my
[7ffff960] 55006370 4e524553 3d454d41 7020796d pc.USERNAME=my p
[7ffff970] 53550063 4f445245 4e49414d 414f525f c.USERDOMAIN_ROA
[7ffff980] 474e494d 464f5250 3d454c49 5450414c MINGPROFILE=LAPT
[7ffff990] 322d504f 434e4435 00505332 52455355 OP-25DNC2SP.USER
[7ffff9a0] 414d4f44 4c3d4e49 4f545041 35322d50 DOMAIN=LAPTOP-25
[7ffff9b0] 32434e44 54005053 433d504d 73555c3a DNC2SP.TMP=C:\Us
```

Memory and registers cleared

SPIM Version 9.1.21 of January 17, 2020
Copyright 1990-2017 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Step 4 : You will get the following Console Window.

