# CS 473 Assignment Three

For this assignment you must create a MIPS assembly language program, and simulate it using the MARS simulator.

The MARS MIPS simulator is available at [http://courses.missouristate.edu/KenVollmar/mars/ (Links to an external site.)Links to an external site.](http://courses.missouristate.edu/KenVollmar/mars/) or at

/home/others1/jcook/public\_html/Mars4\_5.jar

On the CS machines you can run it directly from where it is by just entering the command

java -jar /home/others1/jcook/public\_html/Mars4\_5.jar

There are a variety of ways you could make it a shorter command: create your own shell script, use a shell alias, copy the jar file to where you want, etc., so I'll leave that choice up to you. (**warning**: I haven't found a way to easily shrink the real estate it uses on the screen, so very small laptops might have a problem.)

MARS is an integrated IDE for MIPS assembly code, including an assembler and a simulator.

### Initial MARS Configuration

When you first run MARS, select the "Settings" menu and check the following two settings:

* Initialize Program Counter to global "main" if defined; and
* Program arguments providing to MIPS program

### Starting Out

The assembly code template you can start with is found in [hw3.asm](https://nmsu.instructure.com/courses/1049933/files/102369011/download?wrap=1)View in a new window:. It contains a *main* that has an if-then-else construct and then uses syscall 3 times to print out some stuff, then uses syscall to terminate the program. It also has a stub function *esum* at the bottom, but *main* does not currently call it.

### Assignment Work

You must write the body of the function *esum* so that it produces a sum of *only* the even numbers from the array given as its argument. It must return the sum in register $v0. You must change *main*to correctly call the function *esum* with the appropriate arguments (in registers $a0 and $a1). All values are signed integers. NOTE: You are not allowed to use any pseudo-instructions EXCEPT "la" (load address). In class I have also been using "li" (load immediate), but you should use "ori" instead (with register $zero). Main should then print the value returned in $v0.