

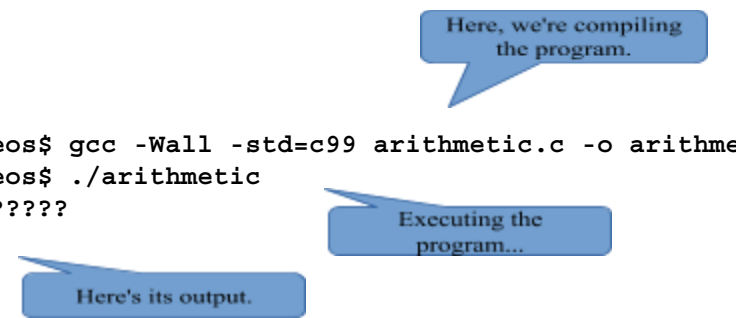
Exercise 01

For this exercise, you get to fill in the missing parts of a simple C program. This will give you a chance to type in some C syntax yourself.

The program you're working on is called `arithmetic.c`; its job is to add up all the numbers from 1 to 10^9 (sum of an arithmetic series from one to one billion). You'll find a shell for this program on the course homepage in Moodle, along with a copy of an expected output file. Download by right-clicking and choosing to save a copy of the file wherever you would like it to be.

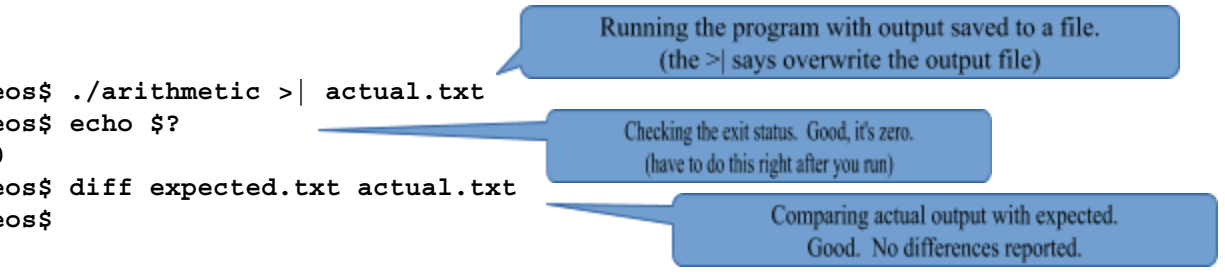
In the source code of `arithmetic.c`, I've given you comments describing the code you need to add. In general, I'll mark places where you need to add code with a comment that looks like `// ...` or like `/* ... */`

Once you've completed the program, you can build and run it with commands like the following, using your choice of platforms (win+cygwin, remote-linux, vcl, macos, etc.).



```
eos$ gcc -Wall -std=c99 arithmetic.c -o arithmetic
eos$ ./arithmetic
?????
```

Make sure your source file is named exactly the right thing (including capitalization), that it compiles cleanly, prints exactly the right output when run and exits with successful exit status. Provided is a sample output file, `expected.txt`, showing exactly what output your program is expected to print. If you put this file in the same directory as your program, you should be able to run commands like the following to check your program's exit status and make sure it prints exactly the right output:



```
eos$ ./arithmetic >| actual.txt
eos$ echo $?
0
eos$ diff expected.txt actual.txt
eos$
```

Once you have completed your program and you're happy with how it runs, submit your `arithmetic.c` source file to the `exercise_01` assignment in Moodle.

BECAUSE THIS IS ASSIGNED LATE, IT WILL NOT BE GRADED FOR CREDIT. A SOLUTION WILL BE POSTED HOWEVER. IT WILL COUNT AS PART OF YOUR "PARTICIPATION" TO SUBMIT, BUT THERE WILL BE NO GRADING.