

Task 2.

Run: module load gcc/latest

- a) Write a C program, task2a.c, that prints (only) the following to standard out without quotation marks: "Hello, world!\n". It will be compiled as follows: gcc task2a.c output.c -o task2a and it will be run as follows: ./task2a
- b) Write a C program, task2b.c, that takes one command line argument, a positive integer n, and calls the outputT2 function from output.h on each integer from 0 up to and including n. It will be compiled as follows: gcc task2b.c output.c -o task2b and it will be run as follows, where n is a positive integer:
./task2b n

A note on the output function: Its goal is to make sure that no one loses points because of a small formatting error. With more complicated output in the future, it will be more necessary than it is here, but we wanted to get you used to the process of using a provided output function now.

- a)
 1. cd /srv/home/cmiao/ME459Upstream/HW04
 2. touch task2a.c
 3. nano task2a.c
 4. input:
#include<stdio.h>

int main(int argc, const char * argv[]) {
 printf("Hello World\n");
 return 0;
}
 5. gcc task2a.c output.c -o task2a

```
[cmiao@euler HW04]$ gcc task2a.c output.c -o task2a
output.c: In function 'outputT5':
output.c:8:5: error: 'for' loop initial declarations are only allowed in C99
e
    for (unsigned int i = 0; i < count; i++) {
    ^
output.c:8:5: note: use option -std=c99 or -std=gnu99 to compile your code
```

6. gcc task2a.c output.c -o task2a -std=c99
7. ./task2a

```
[cmiao@euler HW04]$ ./task2a
Hello World
```

- b)
 1. cd /srv/home/cmiao/ME459Upstream/HW04
 2. touch task2b.c
 3. nano task2b.c
 4. input:
#include <stdio.h>
#include <stdlib.h>
#include "output.h"

int main(int argc, char *argv[]) {
 int i,n=atoi(argv[1]);
 for (i=1;i<=n;i++){
 outputT2(i);
 }
}

5. gcc task2b.c output.c -o task2b -std=c99
6. ./task2b 6

```
[[cmiao@euler HW04]$ ./task2b 6  
1 2 3 4 5 6 [[cmiao@euler HW04]$
```