

# Homework 1

**Due:** April 24, 2023

**Points:** 100

## Gradescope

- (10 points) The goal of this problem is to have you use Gradescope. Please write a program that prints the line “Hello, world!” followed by a newline. Call your program “hello.c”.

## C Programming

- (60 points) You are to write three loops that print numbers in sequence. The sequence may be increasing or decreasing. The file *loopy.c* contains three functions. The first function, *do\_for*, will use a **for** loop; the second, *do\_while*, will use a **while** loop, and the third, *do\_do*, will use a **do ... while** loop. Write each loop within its respective function. Each function has 2 arguments; call the first  $m$  and the second  $n$ .  $m$  is the starting number;  $n$  is the ending number.

If  $m < n$ , then print  $m, m + 1, \dots, n$ , each number on a separate line.

If  $m = n$ , then just print  $m$ .

If  $m > n$ , then print  $m, m - 1, \dots, n$ , each number on a separate line.

The file *loopy.c* is available on the class web site. It contains three functions with the above interface. Write each loop in the appropriate function. Remember to add comments!

Call your program “loopy.c”

## C Program Debugging

- (30 points) On the class web page is the source for a C program named *show.c*. It is supposed to read characters from the standard input and print them on the standard output after expanding any non-printing characters to their C character escape sequence.

The relevant characters, and the C escape sequences to be printed when those characters are encountered, are:

character	print as	character	print as
newline	<code>\n</code>	backslash	<code>\\</code>
horizontal tab	<code>\t</code>	vertical tab	<code>\v</code>
backspace	<code>\b</code>	carriage return	<code>\r</code>
form feed	<code>\f</code>	bell	<code>\a</code>
NUL	<code>\0</code>	<i>anything else</i>	<code>\ooo</code>

The “anything else” entry means that any non-printing character other than the ones named in the table is to be printed as a sequence of three octal digits preceded by a backslash. When the escape sequence for a newline is printed, the program is to skip to the next line.

Unfortunately, the program as saved in *show.c* will not even compile, let alone run. And the programmer thoughtlessly left off all the comments. Hence, your mission: fix the program so it works as described above. You are to turn in a corrected source program, with comments describing the changes you made to get it to work.

Turn in a corrected, fully commented *show.c*.