ME333 Introduction to Mechatronics Name Sean Morton
2022 Quiz 1: C

Turn in your solutions on Canvas. You can answer digitally in a separate pdf or text document, or scan / photo handwritten notes.

1. The following code will overflow. Explain what that means and edit the code to prevent it:

```
unsigned char a = 175; unsigned char b = 100;
unsigned char c = a + b;
```

Every int-like datatype in C has a range of numbers that can be represented by that datatype--for an unsigned char, this range is [0, 255]. Integer overflow is where the value assigned to a variable exceeds the allowable range. Because of the concept of "Two's Complement", when overflow happens the value will wrap around from MAX to MIN: in this example, variable "c" wraps around from 255 to 0, such that even though a + b = 275, variable c = 275 - 256 = 19.

Edited code: unsigned short c = a + b;
range of an unsigned short is [0, 65535] so no overflow

2. In a sentence or two, describe the difference in using

```
int numStates = 3;
```

and

```
#define NUMSTATES 3
```

#define is a preprocessor command and will make our "NUMSTATES" variable a constant--wherever NUMSTATES is written in the code, the preprocessor will replace that variable with value 3.

int numStates is a variable, not a constant, and can be manipulated in the code.

3. Describe what happens during the compilation process when you build a project with multiple .c files

When you build a project with multiple .c files, the compiler turns each .c file into a .o (object) file, and the linker connects each .o file together to make an executable. In order for each .c file to be aware of each other, we need to use header files (.h) and write #include "filename.h" if we want to include functions from filename.c in another .c file. When we do this, the compiler is able to locate the functions from filename.c in memory and paste them into the C files that say #include "filename.h".

Sometimes a header file will be included in more than one other c file in the project. Include guards are there so that no C file can include the same header file more than once.

4. For the following array, state the result if it exists or say “unknown”:

```
unsigned char q[5] = {2, 12, 'j', 128, 10};
unsigned char a; int b;
```

a. `a = *(q+1);` **12**

q[1] = 12

b. `b = ((int)q[3]) << 2;` **512**

q[3] = 128 = 0b 1000 0000.
0b 1000 0000 << 2 = 0b 0010 0000 0000 = 512

c. `a = q[q[0]]+1;` **107**

q[0] = 2
q[2] = 'j' = 106 106 + 1 = 107

d. `b = q[3]*2;` **256**

q[3] = 128
128 * 2 = 256

e. `b = q[4] / q[5];` **unknown**

q[4] = 10
q[5] = out of bounds

5. Use a `while` loop to print all lowercase letters of the alphabet, each on its own line.

6. Create a structure called `car` that contains an integer called `id`, a character array with 50 elements called `brand`, and a float array with 10 elements called `miles`. Write the code that creates an instance of `car`, and initializes the `id` to 1, the first element of `brand` to the null character, and every element of `miles` to 0.0.

7. Write the code that asks the user for the name of a car brand and puts the response in the `brand` field of your answer to #5.

8. Write two functions that produce the same result, one that uses type `car` by value and one that uses `car` by reference. In the functions, calculate the value of each element of `miles` from element 2 to 10, assuming the number of miles doubles from each previous index, and returns to `main` with the calculated values stored inside the variable:

9. Write a complete program (here on the page, you don't have to write or test the code with gcc) that asks the user to enter a string and a number, and then prints the string back with each character shifted over in the ASCII table by the number they entered. For example, if the user enters "ABCD 4", the result is "EFGH". Limit the user to entering a shift value between 1 and 8 (inclusive), keep asking for a shift value until it is in the correct range.