

CS 2263
Assignment #2 Due: Friday October 22nd, 2021

This assignment is due at the start of class on October 22nd (not later) and should be handed in printed as follows.

- These question pages should be printed with your identification added to the top right corner and stapled as the cover page (your assignment must be stapled).
- Questions must be in order but also should always be presented in the following order as separate pages
 - descriptive variable names
 - proper operator spacing (ex: `x = 3` not `x=3`)
 - Proper indentation for all lines of code (use 3 spaces as a suggested tab level)
 - code - make sure it prints properly formatted and indented with headers for the file and for each function. Reprint it if necessary. Printing in 12 point font is preferred. Lines should not “wrap”
 - Compile the output from the suggested outputs (can be edited to be on one page if it fits).
- Programs and output should be printed - not screen capped - and should not be in toner intensive black boxes
- Code should be efficient and readable. Commenting **MUST** appear before any section of code as a brief summary of what is about to happen (think key steps description from your algorithm). Do not code-splain
- Dr. McNally might email and ask for your files if necessary and they should match the printed code exactly.

1. As asked last week, program and hand in question 5 from the first test. Execute the code with the following exact input entered one at a time.

7 30 110 121 190 490 888 500 700 790 888 1021 999

2. As asked last week, program and hand in question 6 from the first test. Write a driver to print out the resulting values for $f(x) = \sin(x)$ for the given intervals and n values in an organized table with each number showing 6 decimal places. Using your knowledge of calculus to include an absolute error column for each case from the true integral

- $a = 0$ $b = \pi/2$, $n = 5$
- $a = 0$ $b = \pi/2$, $n = 10$
- $a = 0$ $b = \pi/2$, $n = 50$
- $a = 0$ $b = \pi/2$, $n = 500$
- $a = 0$ $b = \pi/2$, $n = 5000$
- $a = 0$ $b = 2\pi$, $n = 4$
- $a = 0$ $b = 2\pi$, $n = 8$
- $a = 0$ $b = 2\pi$, $n = 16$
- $a = 0$ $b = 2\pi$, $n = 32$
- $a = 0$ $b = 2\pi$, $n = 64$
- $a = 0$ $b = 2\pi$, $n = 128$
- $a = 0$ $b = 2\pi$, $n = 256$
- $a = 0$ $b = 2\pi$, $n = 512$

3. Design and Code the following:

- (a) Create a driver program which randomly generates a set of 25 ordinal points - (x,y) - where both x and y go from -20 to 20 inclusive. they are to be stored in two separate arrays called **x** and **y**.
- (b) Write a function which prints a point in the format "(x,y)"
- (c) Write a function to print the arrays created earlier using the above function. Call this function from your driver
- (d) Write a function which sorts the arrays provided using increasing order for the x values. Points with similar x values should be then sub sorted into order of y values
- (e) Print the array after having sorted them using the above function

4. Using the features of `string.h`, write a program which will ask a user for their name (assume no longer than 100 characters if necessary) which will be in the format

Last Name, First Name Other Names

such as

McNally, Jefficus Marcus

or

McNally, Dela Grace Hope Aloysius Bartholomew Miracle Magic Marie

Then have your program perform (printing results after each step for verification) the following

- (a) Separate off the family name into a variable called `familyName`
- (b) Separate off the first name into a variable called `firstName`
- (c) Separate off the middle names into a variable called `middleNames`
- (d) Count the number of middle names
- (e) Produce the users name to a variable called **`user`** in normal order (ex: Jefficus Marcus McNally)
- (f) Produce the users name to a variable called **`formalName`** using the first initial and last name with a period (ex. J. McNally)
- (g) Produce the users initials to a variable called **`initials`** (ex. JMM)
- (h) Ask the user for a new last name and replace their old last name with this new one in all the above items.
- (i) Write a function which takes in the information in **`user`** and removes all vowels returning the result (a common facebook hacking question).

Test your program on the two names given, the name “Madonna” , and also your name. In each case replace the family name with something you think is cute.

5. Arrays are static and so they should never change sizes. They should always be full and never have empty spaces but this can be an issue. A `doublingArray` is an attempt to make a dynamic array that can increase its size as needed and also not be full. It starts with a set number of spots (lets assume 2) with no content. As items are added the contents fill in. When a content is added but no space is available, the number of spaces is doubled using memory allocation (`malloc`, `calloc`, `realloc`)
- (a) Define a structure called a `doublingArray` to represent this concept for integers
 - (b) Write a function to create such a structure (think like the `new` operator)
 - (c) Write a function to add an element to the structure
 - (d) Write a function to search the structure for a provided integer
 - (e) Write a function to sort the structure into increasing order
 - (f) Write a function to remove an indicated integer from the structure. You only need remove it once, but it did not have to be in the structure.
 - (g) Write a function to return the memory space of this structure to the computer and manually zero out all the memory spaces for security