CSE 333 25au Exercise 3

out: Wednesday, October 1, 2025
due: Friday, October 3, 2025 by **10:00 am**, **No late exercises accepted.**

Goals: Write a program that uses structs, typedef, and dynamic memory allocation (malloc/free).

Description: Your job is to write a C program that does the following:

- uses typedef to define a new structured type called `Point3d`, which contains `uint16_t` fields for x, y, and z coordinates.
- defines a function called `AllocatePoint3d` that accepts three `uint16_t` values as arguments, mallocs space for a `Point3d`, assigns those three arguments to the x, y, and z fields, and returns (a pointer to) the malloc'ed `Point3d`.
- has a main() that "tests" `Allocate Point3d`. For the purposes of this exercise, it is enough for you to write code that simply allocates a struct by calling Allocate `Point3d` and verifies that the struct members contain the correct values. Your "test" should print an appropriate message if something is wrong; if all is well your program should terminate silently. Make sure your main function frees any memory that `Allocate Point3d` allocates. **Hint:** Don't make assumptions about what `Allocate Point3d` does, even though you implement it. Imagine that your `main` is testing an implementation of `Allocate Point3d` that is implemented by another programmer.

Note: we will be using valgrind to test whether your code has a memory leak! (You probably should too...)

Your code must:

- compile without errors or warnings on CSE Linux machines (lab workstations, attu, or CSE home VM)
- have no crashes, memory leaks, or memory errors on CSE linux machines
- be contained in a single file called `ex3.c` that compiles with the command `gcc -Wall -g -std=c17 -o ex3 ex3.c` -- do not submit a Makefile.
- be pretty: the formatting, modularization, variable and function names, and so on must make us smile rather than cry.
- be robust: you should think about handling bogus input from the user (if any), and you should handle hard-to-handle cases (if there are any) gracefully.
- have a comment at the top of your `.c` file with your name, and CSE or UW email address.

You should submit your exercise using the Gradescope dropbox linked on the course resources web page.