HW08 - Pointers, Pointer Arithmetic & Dynamic Memory [100 pts]

Answer the following questions by modifying the hw08.cpp source file and/or answering the question directly:

1. Complete Q#1 pointer examples. Complete Q#1 reference examples. Indicate if an operation is not allowed and explain why. Complete the implementations for the increment functions (i.e. *increment_value, increment_pointer, increment_reference*). Describe what a pointer is. Describe what a reference is. What's the difference between a pointer and a reference?
2. [read pointer arithmetic, watch pointer arithmetic] Complete Q#2 pointer arithmetic examples. Indicate if an operation is not allowed and explain why.
3. [read arrays and functions, watch pointers and dynamic memory, pointers and 2d arrays] Complete the function definition for the *print_2darray_pointer* function. How is the *twoDDoubles* 2d array laid out in memory? Why is it necessary to cast *twoDDoubles* to a *double** in the *print_2darray_pointer* function call?
4. Complete the dynamic allocation examples. Indicate if an operation is not allowed and explain why. How do the *delete* statements at [4.7], [4.8] affect the values of *variables ri, ri2, ri3*? Add the needed *delete* statement in the function definition for *dynamic_allocation_array_doubles*; this will prevent the function from leaking memory. Add the needed *delete* statement at [4.9] to ensure the *array_of_doubles* returned from the call to *dynamic_allocation_array_doubles* is freed.
5. Complete the function definition for the *print_2darray_dynamic_pointer* function. How is the *p_p_tictactoe* 2d dynamic array laid out in memory (i.e. why is *p_p_tictactoe* an *int***)? Why do we need to pass a

pointer to a pointer of type int in the *print_2darray_dynamic_pointer* function call?

Run *valgrind* to check for memory leaks

Include comments in your code to indicate which code segment answers which question. Appended written answers to the bottom of the hw08.cpp source file (as source comments via //).

Use the command script to capture your interaction compiling and running the program, including all operations, as shown below:

CS1C Spring 2023 TTH HW08 100pts **Due: Tu 3/7/2023**

```
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ script hw08.scr
Script started, file is hw08.scr
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ date
...
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ ls -l
...
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ make all
...
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ ls -l
...
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ ./hw08

... // print out output from steps 1 thru 5

cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ exit
Script done, file is hw08.scr
cs1c@cs1c-VirtualBox ~/cs1c/hw/08 $ make tar
...
Submit the tar package file hw08.tar by Tuesday March 7, 2023.
```