

# CMDA 3634

## Lab 04 Report

Russell J. Hewett

February 10, 2019

### Part A

1. Use the `listings` package to include your output (`output_pt_a_vector.txt`) in your pdf. You will need to copy `output_pt_a_vector.txt` to the reports directory.

**ANSWER:**

```
Test 0: Pass
Test 1: Pass
Test 2: Pass
Test 3: Pass
Test 4: Pass
Test 5: Pass
Test 6: Pass
Test 7: Pass
Test 8: Pass
Test 9: Pass
Test 10: Pass
Test 11: Pass
Test 12: Pass
Test 13: Pass
Test 14: Pass
Test 15: Pass
Test 16: Pass
All Tests: Pass
```

2. For each of the following use-cases, indicate if the specified array should be allocated on the stack, the heap, or either. Explain your selection.
  - (a) An array of integers length 10 in a function that is called a small number of times.
  - (b) An array of doubles of length 3, where  $\sim 10^3$  instances exist and frequently used in the program.
  - (c) An array of doubles of length 3, where  $\sim 10^4$  instances exist and frequently used in the program.
  - (d) An array of doubles of length 3, where  $\sim 10^5$  instances exist and frequently used in the program.
  - (e) An array of doubles of length 3, where  $\sim 10^6$  instances exist and frequently used in the program.
  - (f) An array of doubles of length 3, where  $\sim 10^8$  instances exist and frequently used in the program.
  - (g) An array of floats of length 10,000, to be used throughout the whole program.
  - (h) An array of floats of length 10,000, to be used in a single function.

**ANSWER:**

- (a) You should use the stack because the arrays are small and short lived.

- (b) Either are acceptable, but the stack is probably a better choice if there is risk that the heap will become fragmented.
  - (c) Either are acceptable, but the stack is probably a better choice if there is risk that the heap will become fragmented, as long as the rest of the data used in the program is small.
  - (d) Either are acceptable, but the heap is getting more attractive, as the volume of data is starting to get large.
  - (e) The heap is required, as the volume of data will exceed the available stack memory. Care will need to be taken to avoid fragmentation. This pattern is probably a bad design.
  - (f) The heap is required, as the volume of data will exceed the available stack memory. Care will need to be taken to avoid fragmentation. This pattern is probably a bad design.
  - (g) The heap is preferred, as a single allocation will not fragment, but the stack will be sufficient.
  - (h) The stack is preferred here, as long as the function is not called frequently.
3. In C, there is no mechanism to see if a pointer points to heap memory that has already been allocated, so we cannot be sure that we do not re-allocate an array. How can we code defensively to ensure that this does not happen?

**ANSWER:** We have to be vigilant that we initialize new pointers to `NULL`. Then, we can use custom allocators and deallocators that can check the status of the pointer before acting. As the C language has no support for this sort of protection, which is found in other languages like C++ and Fortran, we are required to code with discipline. Always match allocation to deallocation and always initialize pointers.

4. Other than the instructor or TAs, who did you receive assistance from on this assignment?

**ANSWER:** No one.