Lab 5 - Working with Memory in C

CS120 Spring 2020

**Due Date**  
Midnight, Monday, March 16th

*All programs will be tested on the machines in the G7 lab. If your code does not run on the system in this lab, it is considered non-functioning EVEN IF IT RUNS ON YOUR PERSONAL COMPUTER. Always check that your code runs on the lab machines before submitting.*

# Driver Code and Test Files

* lab5.c
* README.md

# Grading Rubric

**TOTAL: 20 points**

* Part A: *10 points*
  + Each Test Passed: 1-5 (2 points)
* Part B: *8 points*
  + No Memory Leaks or Errors reported by Valgrind
* Part C: *2 points*
  + Follows requested project structure and submission format
  + README is complete

# 

# Guidelines and Policies

## Getting Help

Please follow the debugging guidelines outlined [here](https://docs.google.com/document/d/1gmuVuHQn8rK9p1Clt-SospDACtOAOiy9UPhu-8rPhGg/edit?usp=sharing). We will try to answer questions and provide help within 24 hours of your request. If you do not receive a response in 24 hours, please send the request again.

Although we will answer questions, provide clarification, and give general help where possible up until the deadline, we will not help you debug specific code within 24 hours of the deadline. We will not provide any help after the deadline.

## Guidelines

This is an individual lab assignment. You must do the vast majority of the work on your own. It is permissible to consult with classmates to ask general questions about the assignment, to help discover and fix specific bugs, and to talk about high level approaches in general terms. It is not permissible to give or receive answers or solution details from fellow students.

You may research online for additional resources; however, you may not use code that was written specifically \*to\* solve the problem you have been given, and you may not have anyone else help you write the code or solve the problem. You may use code snippets found online, providing that they are appropriately and clearly cited, within your submitted code.

*By submitting this assignment, you agree that you have followed the above guidelines regarding collaboration and research.*

# Part A

## Dynamic Memory

I have provided you with driver code that explains what you must do for each step. All allocated array memory should be on the heap (using malloc). Below is an overview of the steps:

1. Create an integer array of ***size*** on the heap, and insert -1 into every element
2. Increase the size of your array by 1 element:
   1. allocate space for a new array
   2. copy the values over
   3. make the last value 1, instead of negative 1.
   4. Don’t forget to free the old array.
3. Double the size of your array, copying the values from the first half to the second half
4. Reduce the size of the array by half, only keeping the first half values
5. Remove the first element from the array, and shrink the array by one. The resulting array should be exactly like the previous, only missing the first element.  
     
   All of your loops (printing or otherwise) should use only pointer arithmetic to iterate through your list.

# Part B

## Valgrind

Because C is so flexible, you need to verify that you are using memory properly. Valgrind is a unix memory tool to ensure you are using valid memory (will not work on windows). To get the most out of valgrind, you should use the -g flag when you compile:

|  |
| --- |
| gcc -g myfile.c -o myprog |

Once you’ve compiled your code, you can use valgrind to verify your memory usage:

valgrind ./myprog

Your program will run with additional information. What you are concerned with is the following: lines:

|  |
| --- |
| HEAP SUMMARY:  in use at exit: 10 bytes in 1 blocks  ...  ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0) It should say:  in use at exit: 0 bytes in 0 blocks  0 errors. |

## If it says you have errors or leaks, you will have to go back and try to determine what your error is and fix it.

# Part C

## Submission

Required code organization:

* lab5.c
* README.md

Below is just a reminder of the commands you should use to submit your code. If you cannot remember the exact process, please review Lab 0.

*These commands all presume that your present working directory is within the directory tracked by git.*

You will need to do the following when your submission is ready for grading.

|  |
| --- |
| git commit -a -m "final commit" git push |

To complete your submission, you must copy and paste the commit hash into MyCourses. Go to MyCourses, select CS110, and then assignments. Select this lab, and where it says text submission, paste your commit hash. **DO NOT PASTE ANYTHING OTHER THAN YOUR COMMIT HASH.**    
**Incorrect**: “commit hash: 690fa67ed8”.   
**Correct**: 690fa67ed8

You can get your latest commit hash with the following command:

|  |
| --- |
| git rev-parse HEAD |

Remember, you **MUST** make a submission on MyCourses before the deadline to be considered on time.