Computer Science 301
C Programming in Unix
Siena University
Fall A 2025

Assessment 4: A C Q

Due: 9:00 AM, Wednesday, October 8, 2025

In this assessment, you will build a queue data structure from an existing singly-linked list implementation in C.

You may work alone or in groups of size 2 or 3 on this assessment. However, in order to make sure you learn the material and are well-prepared for the exams, those who work in a group must collaborate closely while completing the assignment (that is, be in the same place at the same time with all group members focused on the task at hand). In particular, the "you do these and I'll do these" approach is sure to leave you unprepared for upcoming tasks and the exams and is prohibited.

This assessment will be graded as a programming assignment. Please review the syllabus description of the expectations for a programming assignment as opposed to a practice program.

You may use AI assistance on this assignment. Such use must be thoroughly documented and credited. The resulting code should be checked carefully to ensure that it satisfies of the design and functionality requirements ane meets course style and documentation expectations.

Learning goals:

1. To gain experience developing a data structure in C by building on an existing structure.

Getting Set Up

In Canvas, you will find a link to follow to set up your GitHub repository, which will be named `acq-assessmentt-yourgitname`, for this assessment. Only one member of the group should follow the link to set up the repository on GitHub, then others should request a link to be granted write access.

All GitHub repositories must be created with all group members having write access and all group member names specified in the `README.md` file by 11:59 PM, Friday, October 3, 2025. This applies to those who choose to work alone as well!

Queue Review

Recall that a queue is a first-in, first-out structure that must support the following operations:

- construction

- enqueue an element
- dequeue an element
- answer whether it is currently empty
- deallocate

From your work in Data Structures, you know that a queue can be built from general purpose linear structures. You probably studied or even implemented a queue using arrays (fixed-size or resizable like an `ArrayList`) or linked lists as building blocks. We will use the singly-linked list with a tail pointer developed in Part 2 of Lab 5 as the building block for a queue.

Question 1: Which of the following is more efficient, and why? Orienting the front of the queue at the head of the linked list, or orienting the front of the queue at the tail of the linked list? (3 points)

Programming Assignment

Your task is to develop a queue data structure in C that holds `int` values, and functions to provide the required queue operations. Include appropriate header files with a structure definition and function prototypes, an implementation file with the function definitions, and a separate file with a `main` function that tests your implementation. You will need to copy your working `sll.c` and `sll.h` from lab into your repository. Also include a `Makefile` that compiles your queue implementation and your testing code.

Important: Don't rewrite all of the linked list code in your queue functions. Your functions should be very short because they should just be making use of the linked list structure and code already provided in `sll.h` and `sll.c`! Your code should not directly access any fields of the `sll` structure.

Submission

Commit and push!

Grading

This assignment will be graded out of 35 points.

Feature	Value	Score
Question 1	3	
Queue operation correctness	20	
Queue test cases	7	
Style and documentation	4	
Working `Makefile`	1	
Total	35	