Assignment 1: Andrew Ortiz and Mitchell Curtis

Problem Description

For this PCB program, we are designing and implementing our very own Process Control Block or, as stated already, PCB. This PCB will take in process objects that contain an ID, Priority, State and stores them into a table for them to organized and added to a ReadyQueue.

Program Design

Describe the overall design of your program.

The overall design of the program utilizes dynamic arrays as the structure to hold the PCB objects into the PCBTable and the ReadyQueue. The program features some max heap implementation that organizes the ReadyQueue by the PCB objects with the highest priority in order to get a very efficient run time of sorting.This algorithm is perfect as were constantly adding and taking away from the ReadyQueue so you need a fast implementation for keeping the top most priority process at the first element of the array.

What algorithms and/or data structures did you choose for implementing the program? Why this algorithm is applicable here? Why do you choose to use those data structures?

For our readyqueue class we used an array of PCB objects that allowed us to add and remove process (PCB) objects dynamically. These objects are added in a max-heap configuration that allows the highest-priority processes to be executed first. The algorithm we used for adding new PCB objects into our already built readyqueue was bubble up. Bubble up is applicable here because it inserts elements into already sorted max-heaps at great efficiency. We chose to use an array of PCB objects as we can initialize it with a very large size and allocate memory manually as needed.

System Implementation

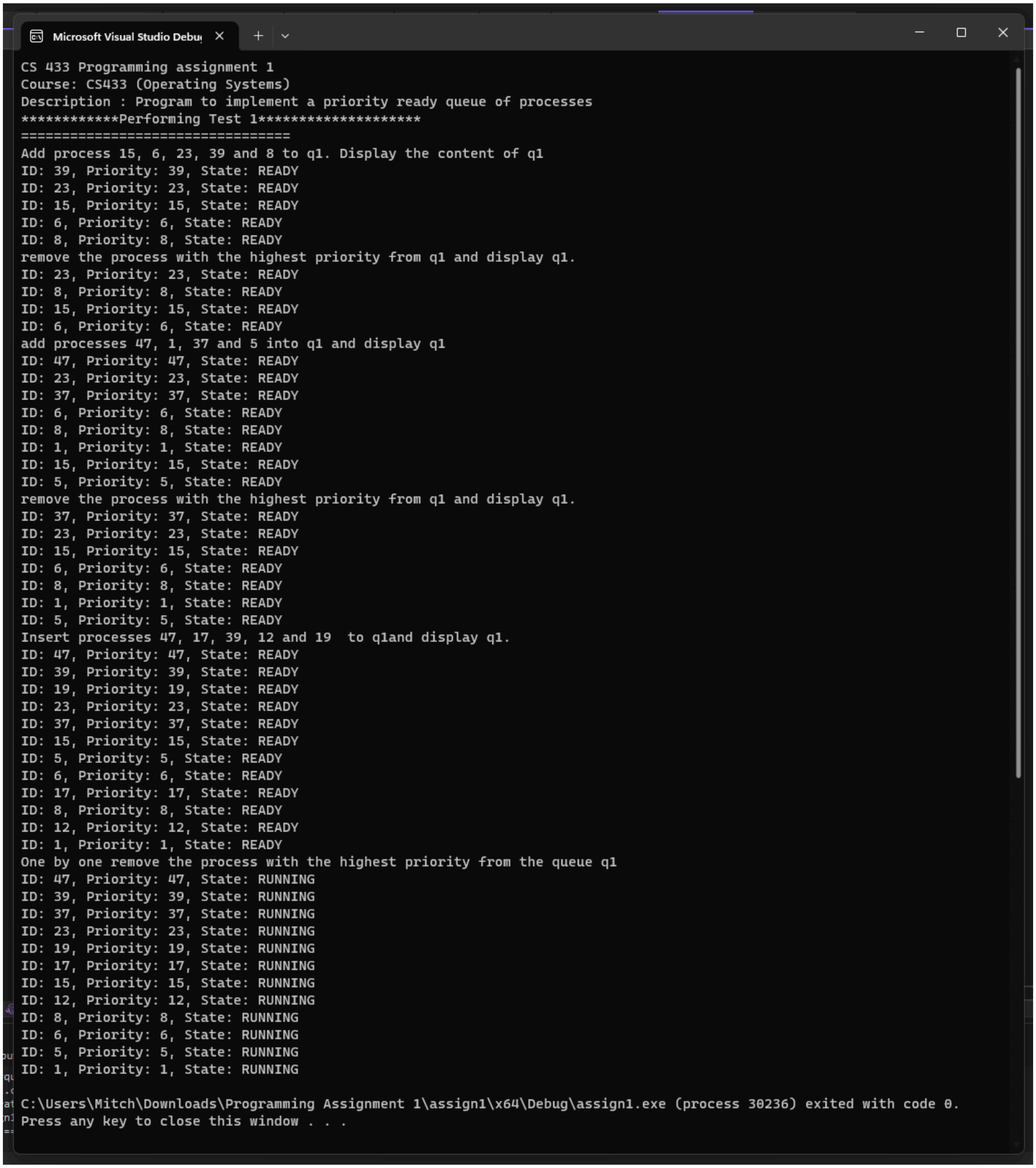
Describe the details of your implementations. Did you run into problems in your implementation? How did you overcome those problems?

We did run into some problems with our implementation, specifically, getting the deconstructor(s) to work as intended. Valgrind would point to issues being within test2.cpp rather than the actual function within the code to where the issue was which made it really hard to debug. I was only able to fix these issues within some of the destructors, while some issues persisted.

Results

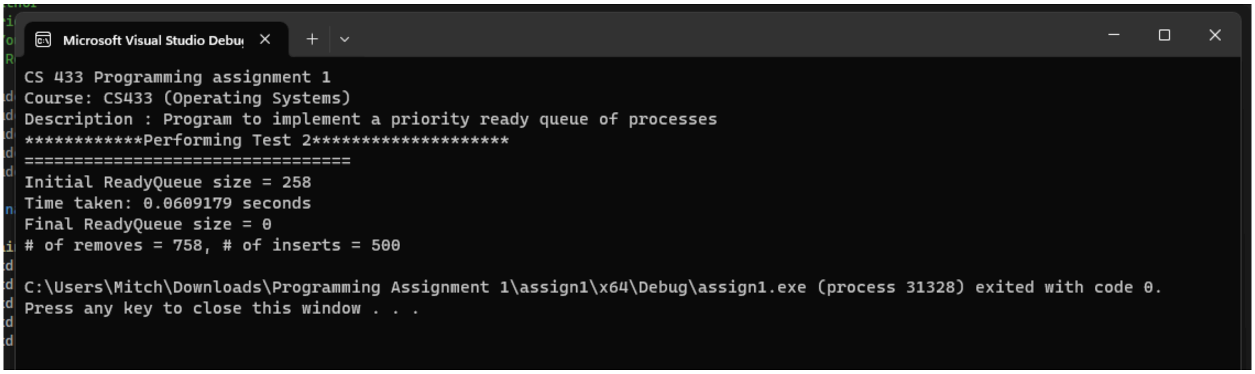
If applicable, your report should include any results for the assignment, the features implemented or missing from the submission. You should give references to the code you used from online/book resources and explain your contributions. You may also discuss extra things you have done in the project, future improvements to your program.

Results:



```
CS 433 Programming assignment 1
Course: CS433 (Operating Systems)
Description : Program to implement a priority ready queue of processes
************Performing Test 1********************
=================================
Add process 15, 6, 23, 39 and 8 to q1. Display the content of q1
ID: 39, Priority: 39, State: READY
ID: 23, Priority: 23, State: READY
ID: 15, Priority: 15, State: READY
ID: 6, Priority: 6, State: READY
ID: 8, Priority: 8, State: READY
remove the process with the highest priority from q1 and display q1.
ID: 23, Priority: 23, State: READY
ID: 8, Priority: 8, State: READY
ID: 15, Priority: 15, State: READY
ID: 6, Priority: 6, State: READY
add processes 47, 1, 37 and 5 into q1 and display q1
ID: 47, Priority: 47, State: READY
ID: 23, Priority: 23, State: READY
ID: 37, Priority: 37, State: READY
ID: 6, Priority: 6, State: READY
ID: 8, Priority: 8, State: READY
ID: 1, Priority: 1, State: READY
ID: 15, Priority: 15, State: READY
ID: 5, Priority: 5, State: READY
remove the process with the highest priority from q1 and display q1.
ID: 37, Priority: 37, State: READY
ID: 23, Priority: 23, State: READY
ID: 15, Priority: 15, State: READY
ID: 6, Priority: 6, State: READY
ID: 8, Priority: 8, State: READY
ID: 1, Priority: 1, State: READY
ID: 5, Priority: 5, State: READY
Insert processes 47, 17, 39, 12 and 19  to q1and display q1.
ID: 47, Priority: 47, State: READY
ID: 39, Priority: 39, State: READY
ID: 19, Priority: 19, State: READY
ID: 23, Priority: 23, State: READY
ID: 37, Priority: 37, State: READY
ID: 15, Priority: 15, State: READY
ID: 5, Priority: 5, State: READY
ID: 6, Priority: 6, State: READY
ID: 17, Priority: 17, State: READY
ID: 8, Priority: 8, State: READY
ID: 12, Priority: 12, State: READY
ID: 1, Priority: 1, State: READY
One by one remove the process with the highest priority from the queue q1
ID: 47, Priority: 47, State: RUNNING
ID: 39, Priority: 39, State: RUNNING
ID: 37, Priority: 37, State: RUNNING
ID: 23, Priority: 23, State: RUNNING
ID: 19, Priority: 19, State: RUNNING
ID: 17, Priority: 17, State: RUNNING
ID: 15, Priority: 15, State: RUNNING
ID: 12, Priority: 12, State: RUNNING
ID: 8, Priority: 8, State: RUNNING
ID: 6, Priority: 6, State: RUNNING
ID: 5, Priority: 5, State: RUNNING
ID: 1, Priority: 1, State: RUNNING

C:\Users\Mitch\Downloads\Programming Assignment 1\assign1\x64\Debug\assign1.exe (process 30236) exited with code 0.
Press any key to close this window . . .
```



```
CS 433 Programming assignment 1
Course: CS433 (Operating Systems)
Description : Program to implement a priority ready queue of processes
************Performing Test 2*******************
=================================
Initial ReadyQueue size = 258
Time taken: 0.0609179 seconds
Final ReadyQueue size = 0
# of removes = 758, # of inserts = 500

C:\Users\Mitch\Downloads\Programming Assignment 1\assign1\x64\Debug\assign1.exe (process 31328) exited with code 0.
Press any key to close this window . . .
```

Conclusion

Give the conclusion and summary of your project. Did it solve the intended problem successfully? What functionalities does your program have? What lessons have you learned from this assignment? Any comments about the project that you want to communicate to the instructor.

While the problem was solved successfully, I still wish I was able to track down the memory leak issues we were having. I’ve learned how PCB’s operate and how their state changes based on what a computer is doing.