

## **PS5:** Ringer Buffer and Guitar Hero

### **PS5a:**

In this assignment, we needed implement the ring buffer that will hold the guitar string position data, and write test functions and exception handling. This assignment was to create the length of the string determining its fundamental frequency of vibration. To accomplish this task, I needed to create a ring buffer that will determine the fundamental frequency and the harmonics of the frequency.

The algorithms we used was a system called the cyclic wrap- around. This concept was to basically reuse an array without extending the amount of elements. I needed to have 2 pointers called the first and last arrays. Each time a user would enter a number in array, the pointer last would place the number in the array and go to the next one. When the user wants to takes out the first array, the first pointer would remove the first number and save the address of the next element. If the queue was full, I could set the pointer last to the beginning of the array. If it was full, it would notify the user its full and stop. Until the user dequeues the the value, the user can not enter any more numbers. If dequeue is continued to be called and the array is empty, the user will be notified that it is empty by throwing a exception.

The understanding of an array that wraps around to replace a element, it a great concept that could save memory in the future. It can be used to be efficient with memory. Whats even more great is that I figured out that members in a class can be accessed with needed gets and sets. Even though I should have known this before, figuring it out in this program helped out a lot. I also learned about throwing a exception when a error has occurred.

```
bara@AL-Laptop: ~/Desktop/ps5a
bara@AL-Laptop:~$ cd Desktop/ps51
bash: cd: Desktop/ps51: No such file or directory
bara@AL-Laptop:~$ cd Desktop/ps5a
bara@AL-Laptop:~/Desktop/ps5a$
bara@AL-Laptop:~/Desktop/ps5a$ make
g++ -Wall -ansi -pedantic -Werror RingBuffer.cpp test.cpp -o ps5a -l boost_unit_
test_framework
g++ -c test.cpp -o test.o
bara@AL-Laptop:~/Desktop/ps5a$ ls
cpplint.py  ps5a      RingBuffer.cpp  RingBuffer.o  test.o
Makefile    ps5a-readme.txt RingBuffer.hpp   test.cpp
bara@AL-Laptop:~/Desktop/ps5a$ ./ps5a
Running 3 test cases...
0
100
1
100
2
100
0
3
1
3
2
3
3
3
3
3
3
3
*** No errors detected
bara@AL-Laptop:~/Desktop/ps5a$
```

(Compiled with Makefile)