**Planning Document – Franklyn Gonzalez P5 Program: Procrastinate**

**Planning Phase**

Prior to working on my programming assignment, I plan to create a struct that lists all the items that associate with each task. Such as the id number, time assigned, number of phases, integer array, total time left, and the number of which task to complete next. I want to have a function that allows me to initialize variables inside the struct and add this into my number of tasks.

After I have this set up, I plan to run a simulation that will allow me to: initialize my binary heap, remove and return the top element off the heap, and add any tasks that were given while working on the previous phase.

**Assistance Received**

I had no assistance on the assignment.

**Debugging Phase**

During the debugging phase, I had a problem understanding what was required with the ID mentioned in P5. A lower ID meant a task would have came before another in the ordering. If I did this, two lines of output would have been backwards from what the assignment expected.

For certain sections, I had to make sure that I was able to free the heap, task, phaselength, and currentTask before submitting this assignment.

**Testing Phase**

I had written a heap-printing function that took a pointer to a BinaryHeap as an input, and displayed the entire tree in a neatly, formatted way, with tabs to organize levels. This way, I was able to make sure that the heap implementation was correct by trying to add values (time, number of phases) to it and made sure it responded as expected.

With that being said, I built the project on codeblocks with the testcase provided below…

5  
10 3 180 3000 2000  
200 5 1800 5000 7000 1000 60  
1000 2 20000 1  
2000 8 1000 1000 1000 1000 1000 1000 1000 1000  
1000000 6 10 20 30 30 20 10  
  
And printed as expected on the sample output.

