Brian Huang
CS162
2/26/2016

Reflection

I had the most difficulty with this homework assignment than any of the others. This is because I did not really understand lab6, the linked list lab fully so I struggled to complete this. To complete this assignment I had research how to use linked lists on the internet. Even after the research I still don't quite understand linked lists fully.

To create this homework I created Stack, Stacknode, Queue, Queuenode classes to work with my homework3. I also modified the fight() function from homework3 and changed it into tournament(). This function worked exactly the same as the fight function except that it would have multiple monsters fighting, and it would loop through the teams. To determine the winner, I first dumped all the remaining living creatures into the loser stack. The winning team was simply the team that still had living creatures on it. The first, second and third place winners are determined by who entered the losers stack last.I did it this way because it was the easiest to implement.

Testing for this homework was a bit more difficult than the others. I first tested if my creation of the creatures were working, so I created a linked list of size 1 and had them fight. After that I worked on creating the looping feature that would make multiple creatures fight. Creating the winner pile was by far the most complicated to test. The way I tested it was to display the last winner, then I worked backwards to display the rest of the loser pile. My display function did not work properly at first. I ran into multiple cases of segfaults, but I soon realized that I was allocating data that was no longer there. After I fixed that problem, the display feature works fine.

Most of the ways I resolved problems for this program was to make the information be outputted and displayed. This way I can see if the creatures are actually looping, or if the indices are actually working, and if the creatures are healing. I also added a feature that asks the user if they want to continue every round. This initially was a debug feature to run through the program slowly, and check the numbers. After I finished debugging I decided to keep this feature. I thought it made using my program easier since it breaks up the code into smaller chunks for the users to read.