CS130 - LAB 3 - Debugging Problems

Name: Stephanie Cabrera **SID:** 861288909

The codes we will be working with can be found on iLearn.

1. Run **valgrind** on prog-1.
   a. What type of error do we get and why?

   Memory leak w/ 20,604 bytes in 101 blocks lost. calling memory that is not there data[].

   b. How prog-1 can be changed so we don't get this error anymore?

   In resize, delete data[] after the for loop and before setting the data pointer to new data[]. In append make Temp equal to parameter then make data[i] = temp.

2. Run **valgrind** on prog-2.
   a. What type of error do we get and why?

   10 bytes of lost memory

   Reachable memory → not being called correctly

   the function setname doesn't know what object type the objects is.

   b. How prog-1 can be changed so we don't get this error anymore?

   overload the setname operator for each object type: sphere, points

3. Run **gdb** on prog-3.
   a. The program should stop with segmentation fault exception. Type **list** to see the region where the program stopped. In which line of code is the program crashing?

   line 10

   b. Use the command **print** <statement> with the variables that are being accessed on the line where the program is crashing. Why the program crashed in this case and how we can fix it?

   size_t at 0 is fine. but when size_t -1 happens, it sets back to size_t = max value

4. Run **valgrind** on prog-4.
   a. What type of error do we get and why?

   Out of bound error because arr[] should have a null at the end to indicate that the array string ended.

   b. How prog-4 can be changed so we don't get this error anymore?

   Change the char arr[] to string because it is already implemented.

5. Run **gdb** on prog-5 and follow the steps below.

   a. The program should stop with an segmentation fault exception. In which line of code is the program crashing?

   When it calls n->next->prev because n->next = null when n = tail.

   b. Why the program crashed in this case and how we can fix it? (you may want to see the *list* and *node* structures in the source code for this)

   It crashes when adding a node after the tail because it doesn't check for it

   c. Compile and run the program again using **gdb**. The program should crash again. Try using **list** and **print** to figure out why the program is crashing and briefly explain your reasoning. What changes need to be made in the code to fix this problem?

   The code doesn't account for removing the head or tail nodes only if the node is in the middle of head and tail.

   d. Compile and run the program again using **valgrind**. The program should display an error. Why do we get this error and how we can fix it?

   Write if conditions for removing head, tail, and when head == tail.

6. Using **gdb** and **valgrind** (use the best for each situation), briefly describe all problems in prog-6 and propose fixes for each one of the problems.

Overload operators correctly,
including a deconstructor for deallocating memory.