Project 1 Focused on basic C programming skills

- Dynamic memory management with malloc()/free()
- Reading data from files in text format
- Displaying information to the screen
- Reading commands from users in interactive programs
- Building data structures with C structs

This project had 2 different programs

1) A Stock DataBase
2) Hashtable

We were responsible for make the .c classes based off of the given header files.

Stock

```c
// stock_funcs.c
void stock_print(stock_t *stock);
stock_t *stock_new();
void stock_free(stock_t *stock);
void stock_set_hilo(stock_t *stock);
int stock_set_best(stock_t *stock);
int count_lines(char *filename);
int stock_load(stock_t *stock, char *filename);
void stock_plot(stock_t *stock, int max_width, int start, int stop);
```

HashTable

```
// functions defined in hash_funcs.c
long  hashcode(char key[]);
int   next_prime(int num);

void  hashmap_init(hashmap_t *hm, int table_size);
int   hashmap_put(hashmap_t *hm, char key[], char value[]);
void  hashmap_expand(hashmap_t *hm);
char *hashmap_get(hashmap_t *hm, char key[]);
void  hashmap_free_table(hashmap_t *hm);

void  hashmap_write_items(hashmap_t *hm, FILE *out);
void  hashmap_show_structure(hashmap_t *hm);
void  hashmap_save(hashmap_t *hm, char *filename);
int   hashmap_load(hashmap_t *hm, char *filename);
```

Gradescope Testing

PROBLEM 1 (15.0/15.0)

```
gcc -Wall -Wno-comment -Werror -g  -c stock_funcs.c
gcc -Wall -Wno-comment -Werror -g  -o test_stock_funcs test_stock_funcs.c stock_funcs.o
./testy test_stock1.org
============================================================
== test_stock1.org : Problem 1 First 3 Functions in stock_funcs.c
== Running 15 / 15 tests
1)  stock_new            : ok
2)  stock_free1          : ok
3)  stock_free2          : ok
4)  stock_free3          : ok
5)  stock_free4          : ok
6)  stock_print1         : ok
7)  stock_print2         : ok
8)  stock_print3         : ok
9)  stock_print4         : ok
10) stock_print5         : ok
11) stock_print_prices_0 : ok
12) stock_print_prices_1 : ok
13) stock_print_prices_2 : ok
14) stock_print_prices_3 : ok
15) stock_print_final    : ok
============================================================
RESULTS: 15 / 15 tests passed
```

PROBLEM 2 (15.0/15.0)

```
gcc -Wall -Wno-comment -Werror -g  -c stock_main.c
gcc -Wall -Wno-comment -Werror -g  -o stock_main stock_main.o stock_funcs.o
./testy test_stock2.org
============================================================
== test_stock2.org : Problem 2 Remaining Functions in stock_funcs.c
== Running 15 / 15 tests
1)  stock_set_hilo1         : ok
2)  stock_set_hilo2         : ok
3)  stock_set_hilo3         : ok
4)  stock_set_best1         : ok
5)  stock_set_best2 and 3   : ok
6)  stock_set_best4         : ok
7)  count_lines             : ok
8)  stock_load1             : ok
9)  stock_load2 and 3       : ok
10) stock_load pathological : ok
11) stock_plot1             : ok
12) stock_plot2 3 4         : ok
13) stock_plot5 6           : ok
14) stock_main1             : ok
15) stock_main2             : ok
============================================================
RESULTS: 15 / 15 tests passed
```

PROBLEM 3 (20.0/20.0)

```
gcc -Wall -Wno-comment -Werror -g  -c hashmap_main.c
gcc -Wall -Wno-comment -Werror -g  -c hashmap_funcs.c
gcc -Wall -Wno-comment -Werror -g  -o hashmap_main hashmap_main.o hashmap_funcs.o
./testy test_hashmap.org
============================================================
== test_hashmap.org : Problem 3 hashmap_main application
== Running 20 / 20 tests
1)  start, print, quit          : ok
2)  hashcode and get w/ empty   : ok
3)  put single and get          : ok
4)  put 3 key/vals              : ok
5)  put with collisions         : ok
6)  check -echo option          : ok
7)  put 3 and show structure    : ok
8)  put overwrites              : ok
9)  Larger Hash with Structure  : ok
10) End of File EOF             : ok
11) clear Command               : ok
12) save command, 3 puts        : ok
13) save command, many puts     : ok
14) load existing file          : ok
15) load file with larger array : ok
16) load fails                  : ok
17) saving and loading          : ok
18) next_prime command          : ok
19) expand command              : ok
20) stress testing              : ok
============================================================
RESULTS: 20 / 20 tests passed
```