Pony Tracker – Midterm Project Report

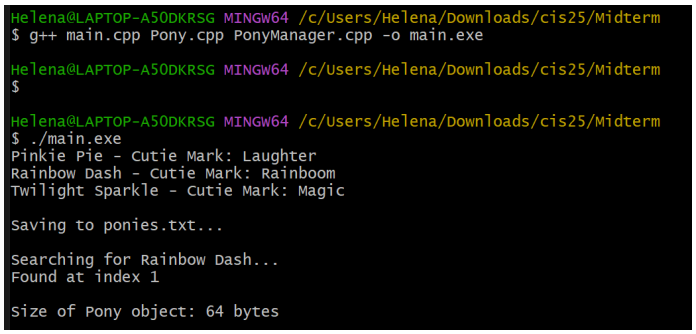
What My Program Does

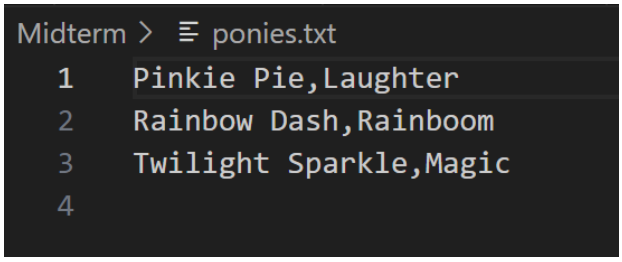
My program is a simple pony tracking system. It lets you store and manage a list of My Little Ponies along with their cutie marks. You can add new ponies, display the list of ponies, sort them by name, search for a specific pony using binary search, and save or load pony data from a file. It uses classes to keep everything organized, and the ponies are stored using a dynamically allocated array of pointers.

Concepts I Used

This project uses several core programming concepts. I used basic data types like string and int, and I used the sizeof() operator to display memory size. I handled text files for reading and writing pony data. The ponies are stored using pointers in an array, which requires dynamic memory management. I implemented a binary search function to find ponies by name, and used a basic sort function to make sure the search worked correctly. The project makes use of std::string for storing names and cutie marks. Finally, I created two interacting classes: one for individual ponies and another for managing the collection.

Screenshots





Challenges Faced

One of the first issues I ran into was a linker error saying it couldn't find a reference to Item::Item(...). This happened because I forgot to compile the .cpp file where I defined the class methods. I fixed it by making sure I compiled all the necessary .cpp files together.

Another issue was that binary search wasn’t working at first. I realized I was trying to search an unsorted array, so I added a sorting step before calling the search function.

I also had problems with my development environment. When I tried compiling with MSYS2, I got an error about WinMain. I don’t know how but it turns out I was using the wrong shell (UCRT64 instead of MINGW64), and I wasn’t linking the program correctly. I fixed this by switching to the correct shell and making sure my main() function was set up properly.

Lastly, I initially stored ponies as normal objects, but that didn’t work well with dynamic memory. I ended up switching to using new to create each pony and storing them as pointers, which gave me more control.