1. Jorrin Thacker - jct13761@uga.edu
2. List of the functions I implemented:
    a. file_read() - This functions reads the file starting at the offset and continues for the size specified. It has several tests to make sure that all is valid, including: file does not exist, file is a directory, and incorrect/disproportionate size for the file specified. I have tested all these catches and if they are not met, the file correctly reads the size specified starting from the offset.
    b. file_remove() - This function removes the specified file from the directory and frees up the i-node and data blocks that were previously occupied by the file. It has several tests to make sure that all is valid, including: file does not exist, file is a directory, and if the file has any associated hard links to it. If there are hard links, it simply removes the file from the directory list and decrements the file count. If there are no hard links, it will remove the name from the directory list, free the i-node number in the i-node bitmap, and finally free all the blocks associated with it in the block bitmap.
    c. hard_link() - This function creates a hard link for the src file specified. It has several tests to make sure that all is valid, including: src file does not exist, src file is a directory, dest file exists, directory is full, and data/i-node blocks are full. If none of these conditions are met, it will then update the last access time of the src file, increment the link_count variable, and finally add the new hard link file to the directory with the same information (data block, i-node number) as the src file.
    d. dir_make() - This function creates a new directory and adds it to the directory list. It has several tests to make sure that all is valid, including: directory exists, current directory is full, and data/i-node blocks are full. If all these pass, it will then set the new i-node data and write it to the current directory. It then creates a new Dentry object to hold the info for the new directory like the "." and ".." directories, then writes this object to the directory's data block.
    e. dir_remove() - This function removes a directory from the current directory. It has several tests to make sure that all is valid, including: directory does not exist, and directory is a file. If these tests pass, then it reads the Dentry object from the directory’s block and uses its info to determine if the directory to be deleted is either the current or parent directories or if the directory is empty, which will throw an error. If all these pass, then the directory is removed from the current directory and the i-node and data bitmaps are updated to be empty at the directory's position.
    f. dir_change() - This function changes the current directory into the specified directory. It has several tests to make sure that all is valid, including: directory does not exist and directory is a file. If both these pass, then it writes the current directory list to its data block, sets the current block to the new directory's, and reads the new directory's list into memory.
3. My design for this project was to look at the functions that had already been implemented and to try to use them as a starting point for the functions I implemented. Most of the design of this project was already implemented in the started code, so it was more implementing the functions than worrying too much about design. The most difficult to get correctly working was the dir_create() as it was difficult to figure out what needed to be different from file_create() and get it working as intended.