Csci 360 lab8 report Kevin Skinner

In part A I used a struct containing an array of size 10. This was used as my inode that could hold 10 blocks of info as it has no indirection. In part B I used a struct containing a char * and a struct file array of size 10. This struct was used to hold the information of the directory by using the char * for the name and a struct file array to hold the files in the directory. Also I used a struct containing a struct inode array of size 10. This was to hold the different inodes(files) within the directory. For free space management I used an array of size 100 that contained all zeros and was checked if a requested random block was taken if the block was not taken it was set to one and was taken else a new block was randomly requested. To extend the design to include a single indirection block I would make a check to see if the inode block is the 10th block and if so I would get a new inode get the 10th block to point to the new inode. Also, I would get new blocks from free space management.

1)first of all my files don't have names and are just the inodes so I would add to my file struct a char * file_name then check which file is to be renamed based on its directory and file name. Then get the user to enter a new name for the file and store it in the location of where the old file name was.

2)first find the file based on directory name and file name then delete the file by removing the inodes that are part of that file and setting the free space management array positions used by the inodes

3)   I would use a triple indirection block and offset 125 would be located in the triple indirection block at the 1th position in the first indirect 1th position in the second indirect and 5th position in the third indirect