```c
// Nate McCain
// CS 390
// Project 4
// ProjectFourFunctions.c
// 10/25/2017

#include "ProjectFourFunctions.h"

// This is to append a forward slash to a pathname.
char forwardSlash[1] = {"/"};

// Format the user's input into a directory path by putting a forward
// slash at the end. Put the result in directoryName.
void formatToDirectoryName(char *directoryName, const char *userInput)
{
    strcpy(directoryName, userInput);
    strcat(directoryName, forwardSlash);
}

// Take the user's input and make it into a directory filepath. Then append the
// latest directory entry to the pathname.
void formatToPathname(char *pathname, const char *userInput, char *directoryEntry)
{
    formatToDirectoryName(pathname, userInput);
    strcat(pathname, directoryEntry);
}

// Record statistics of the status record.
void takeStatistics(int *directoryCount, int *regularCount, long long *byteCount,
 long long *blockCount, struct stat statHolder)
{
    if (S_ISDIR(statHolder.st_mode) == 1)
    {
        *directoryCount += 1;
    }

    if (S_ISREG(statHolder.st_mode) == 1)
    {
        *regularCount += 1;
    }

    *byteCount += statHolder.st_size;
    *blockCount += statHolder.st_blocks;

}

// Write the final statistics to the output file.
 void writeStatistics(int directoryCount, int regularCount, long long byteCount,
  long long blockCount, FILE *fileToWrite)
{
    fprintf(fileToWrite, "The number of regular files is: %d \n", regularCount);
    fprintf(fileToWrite, "The number of directory files is: %d \n", directoryCount);
    fprintf(fileToWrite, "The total number of bytes required for the entire
     directory is: %lld \n", byteCount);
```

```c
    fprintf(fileToWrite, "The total number of blocks required for the entire
     directory is: %lld \n", blockCount);
}

// Read the final statistics of the output file.
 void outputStatistics(FILE *file)
{
    int i = 0; // Counter for the while loop.
    char stuffToOutput[100]; // Holds what is written in the file.
    while (i < 4)
    {
        // Get the next line in the output file.
        fgets(stuffToOutput, 150, file);
        // Echo each line to the terminal.
        printf("%s \n", stuffToOutput);
        // Increment the counter.
        i += 1;
    }
}
```