```
/* Compare Aaron's music & new music files
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>         // for exit()
#include <errno.h>
#include <sys/stat.h>

#define BUFFER 1024         // Maximun length of input line -- probably < 80 chars

long unsigned getFilesize(const char* filename);

int main(int argc, char *argv[])
{
    FILE *fHndl01;          // File handle for first music file
    FILE *fHndl02;          // File handle for first music file
    char  line01[BUFFER];
    char  line02[BUFFER];
    int   nLenLine01 = 0;
    int   nLenLine02 = 0;
    int   nNew     = 0;
    int   nOld     = 0;
    long unsigned nSize01 = 0;
    long unsigned nSize02 = 0;
    long unsigned nTemp   = 0;
    long unsigned nPos    = 0;

    if(argc != 3)
    {
        fprintf(stderr, "\nUsage: Compare fName01 fName02\n"
                        "         Where fName01 & fName02 are music files\n");
        exit(EXIT_FAILURE);
    }

    // See if the files exist and pick the one with the largest size
    nSize01 = getFilesize(argv[1]);
    nSize02 = getFilesize(argv[2]);

    // Check to see which is greater or if an error occured
    if(nSize01 == 0)
    {
        fprintf(stderr, "File: %s not found\n", argv[1]);
    }
    if(nSize02 == 0)
    {
        fprintf(stderr, "File: %s not found\n", argv[2]);
    }

    if( (nSize01 == 0) || (nSize02 == 0) )
    {
        fprintf(stderr, "Error: can't continue until both files can be opened\n");
        exit(EXIT_FAILURE);
    }

    // Open the files - largest file as fHndl01
    if( nSize01 > nSize02)
    {
        printf("Size01, Size02: %lu, %lu\n", nSize01, nSize02);
        fHndl01 = fopen(argv[1], "r");
        fHndl02 = fopen(argv[2], "r");
    }
    else
    {
        printf("Size01, Size02: %lu, %lu\n", nSize01, nSize02);
```

```
        nTemp   = nSize02;
        nSize02 = nSize01;
        nSize01 = nTemp;

        fHndl01 = fopen(argv[2], "r");
        fHndl02 = fopen(argv[1], "r");
    }

    if( (fHndl01 == NULL) || (fHndl02 == NULL) )
    {
        fprintf(stderr, "Error: couldn't open file %s\n", argv[2]);
        exit(EXIT_FAILURE);
    }

    // Read file until no more text
    system("cls");
    while(fgets(line02, BUFFER, fHndl02) != NULL)       // Read from smaller file on outside loop
    {
        // Read a line from the smaller file
        nLenLine02 = strlen(line02);
        if(nLenLine02 > 1) line02[nLenLine02 - 1] = '\0';       // Remove \n

        // Get File Position of fHndl02
        nPos = ftell(fHndl02);
        while(fgets(line01, BUFFER, fHndl01) != NULL)
        {

            nLenLine01 = strlen(line01);
            if(nLenLine01 > 1) line01[nLenLine01 - 1] = '\0';       // Remove \n

            //printf("%3d %s\n", nLenLine01, line01);

            // See if line02 matches line01, if not its a new album from the old
            if(strncmp(line01, line02, nLenLine02) != 0)
            {
                // New album
                ++nNew;
                printf("New %s\n", line01);
            }
            else
            {
                // Old album
                ++nOld;
                printf("    %s\n", line01);
                break;
            }
        }

        if(strlen(line01) == 0)
        {
            fprintf(stderr, "Ran through the new (longer) file\n");
            fseek(fHndl01, nPos, SEEK_SET);
            exit(EXIT_FAILURE);
        }
    }

    printf("Number new albums: %3d\n", nNew);
    printf("Number old albums: %3d\n", nOld);
    printf("Total  all albums: %3d\n", nOld+nNew);

    fclose(fHndl01);
    fclose(fHndl02);

    exit(EXIT_SUCCESS);
}
```

```
/**
 * Get the size of a file.
 * @return The filesize, or 0 if the file does not exist.
 */
long unsigned getFilesize(const char* filename)
{
    struct stat st;

    errno = 0;
    if(stat(filename, &st) != 0) {
        return 0;
    }
    return st.st_size;
}
```