```
/* 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
#define CommonFile "CommonFile.txt" // Temp file to store albums in both old & new music files

// Functions
int  InitParams(int argc, char *NewFile, char *OldFile);
int  MakeCommonAlbum(void);
long unsigned getFilesize(const char* filename);
void CreateComFile(void);

char  AlbumNew[BUFFER];
char  AlbumOld[BUFFER];
char  AlbumCom[BUFFER];

FILE *hNewFile = NULL;          // File handle for new music file
FILE *hOldFile = NULL;          // File handle for old music file
FILE *hComFile = NULL;

int main(int argc, char *argv[1])
{
    int   nLenNew = 0;
    int   nLenCom = 0;
    //  int   nLenOld = 0;
    int   nNew    = 0;
    int   nOld    = 0;
    int   nCommon = 0;
    //  int   nSkip   = 0;
    //  int   nErrs   = 0;
    int   nTotNew = 0;
    //  long unsigned nPosNew  = 0;
    //  long unsigned nPosOld  = 0;

    // Check for proper usage, Initialize files & create temp file for common albums in old & new
    nTotNew = InitParams(argc, argv[1], argv[2]);

    CreateComFile();
    nCommon = MakeCommonAlbum();

    fprintf(stderr, "Number new albums:%5d\n", nTotNew);
    fprintf(stderr, "Number old albums:%5d\n", nCommon);
    fprintf(stderr, "Number dif albums:%5d\n", nTotNew - nCommon);

    if(hNewFile == NULL) printf("hNewFile is NULL\n");
    if(hOldFile == NULL) printf("hOldFile is NULL\n");
    if(hComFile == NULL) printf("hComFile is NULL\n");

    fprintf(stderr, "Beginning of program after calling InitParams, CreateComFile &
    MakeCommonAlbum\n");
    fprintf(stderr, "Press [Enter] key to continue");
    getchar();

    // Find the differences between the common albums and the new albums
    // Open common file

    rewind(hNewFile);
    rewind(hOldFile);
    rewind(hComFile);
    fgets(AlbumNew, BUFFER, hNewFile);
    fgets(AlbumCom, BUFFER, hComFile);
    fprintf(stderr, "New File: %s", AlbumNew);
    fprintf(stderr, "Tmp File: %s", AlbumCom);
    fprintf(stderr, "\n--- After reading ----");
    rewind(hNewFile);
    rewind(hComFile);
    fprintf(stderr, "Press [Enter] to continue\n");
    getchar();

    nOld = nNew = nTotNew = 0;
    fprintf(stderr, "Starting Main While Loop\n");
    while(fgets(AlbumCom, BUFFER, hComFile) != NULL)
    {
        nLenCom = strlen(AlbumCom);
        if(nLenCom > 2) AlbumCom[nLenCom -1] = '\0';        // Remove '\n'

        ++nTotNew;
        while(fgets(AlbumNew, BUFFER, hNewFile) != 0)
        {
            if(nTotNew < 4) fprintf(stderr, "Inner Loop %s\n%s\n", AlbumNew, AlbumCom);
            nLenNew = strlen(AlbumNew);
            if(nLenNew > 2) AlbumNew[nLenNew -1] = '\0';
            if(strncmp(AlbumCom, AlbumNew, nLenNew) == 0){
                ++nOld;
                break;
            }
            else{
                ++nNew;
                printf("%s\n", AlbumNew);
            }
        }
        if(feof(hNewFile)){
            fseek(hNewFile, 0, SEEK_SET);
        }
    }

    fprintf(stderr, "Number all albums:%5d\n", nTotNew);
    fprintf(stderr, "Number old albums:%5d\n", nOld);
    fprintf(stderr, "Number new albums:%5d\n", nNew);

    fclose(hNewFile);
    fclose(hOldFile);
    fclose(hComFile);

    return 0;
}


/*-------------------
* Check proper usage
*-------------------
*/
int InitParams(int argc, char *NewFile, char *OldFile)
{
    int  nErrs   = 0;
    int  nTotNew = 0;
    long unsigned nSizeNew = 0;      // File Size New Albums file
    long unsigned nSizeOld = 0;      // File Size Old Albums file
    long unsigned nTemp    = 0;

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

```c
    // See if the files exist and pick the one with the largest size
    nSizeNew = getFilesize(NewFile);
    nSizeOld = getFilesize(OldFile);

    // Check to see which is greater or if an error occured
    if(nSizeNew == 0) {
        ++nErrs;
        fprintf(stderr, "File: %s not found\n", NewFile);
    }
    if(nSizeOld == 0) {
        ++nErrs;
        fprintf(stderr, "File: %s not found\n", OldFile);
    }

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

    // Open the files - largest file as hNewFile
    nErrs = 0;
    if( nSizeNew > nSizeOld) {
        if( (hNewFile = fopen(NewFile, "r")) == NULL) ++nErrs;
        if( (hOldFile = fopen(OldFile, "r")) == NULL) ++nErrs;
    }
    else {      // Switch files so that the largest one is called new music file
        nTemp    = nSizeOld;
        nSizeOld = nSizeNew;
        nSizeNew = nTemp;

        if( (hNewFile = fopen(OldFile, "r")) == NULL) ++nErrs;
        if( (hOldFile = fopen(NewFile, "r")) == NULL) ++nErrs;
    }

    if( nErrs ) {
        fprintf(stderr, "Error: couldn't open file %s or %s\n", NewFile, OldFile);
        exit(EXIT_FAILURE);
    }

    fprintf(stderr, "New File size: %lu, Old File Size %lu\n", nSizeNew, nSizeOld)  ;

    // Read New File to find the number of new albums
    while(fgets(AlbumNew, BUFFER, hNewFile) != NULL)
    {
        ++nTotNew;
    }

    rewind(hNewFile);           // Rewind New Music file

    return nTotNew;

}


/* ------------------
* CreateComFile()
*
* ------------------
*/
void CreateComFile(void)
{
    // Create temp file to store albums that are in both the old & new albums files
    if( (hComFile = fopen(CommonFile, "w+")) == NULL){
        fprintf(stderr, "Can't open %s for writing\n", CommonFile);
        exit(EXIT_FAILURE);
    }
}


/* ----------------------
* Create file with all the common albums in itoa
*
* ---------------------
*/
int MakeCommonAlbum(void)
{
    int   nLenNew = 0;
    int   nLenOld = 0;
    int   nOld    = 0;
    int   nWrote  = 0;
    fprintf(stderr, "Entering MakeCommonAlbum()\n") ;
    // Create temp file containig albums common to both files
    fseek(hOldFile, 0, SEEK_SET);
    fseek(hNewFile, 0, SEEK_SET);

    while(fgets(AlbumOld, BUFFER, hOldFile) != NULL)        // Read old music file on outside
    loop
    {
        // Now we have an album from the old music file
        nLenOld = strlen(AlbumOld);

        // File Position of new list which is pointed to by hNewFile
        while(fgets(AlbumNew, BUFFER, hNewFile) != NULL)
        {

            // Now we have an album from the new music list
            nLenNew = strlen(AlbumNew);

            // See if AlbumOld matches AlbumNew, if not its a new album from the old
            if(strncmp(AlbumNew, AlbumOld, nLenNew) == 0) {
                // Common album
                errno = 0;
                fprintf(hComFile, "%s", AlbumNew);
                if(errno){
                    fprintf(stderr, "AlbumNew:%3d %s\n", nLenNew, AlbumNew);
                    fprintf(stderr, "AlbumOld:%3d %s\n", nLenOld, AlbumOld);
                    fprintf(stderr, "Number written: %3d  Error Number: %3d\n", nWrote, errno);
                    fprintf(stderr, "Error as string: %s\n", strerror(errno));
                    fprintf(stderr, "File handle: %p\n", hComFile);
                    exit(EXIT_FAILURE);
                }
                ++nOld;
                break;
            }
        }

        if(feof(hNewFile)) {                // Went through all new Albums didn't find old one
            fseek(hNewFile, 0, SEEK_SET);
        }
    }
    fflush(hComFile);           // Keep hComFile open but flush the data
    fprintf(stderr, "Finished MakeCommonAlbum, Number record: %4d\n", nOld);
    fprintf(stderr, "Press [Enter] to continue\n");
    getchar();

    return nOld;
}
```