```
/*------------------------------------------------ VerifyFiles.c -----
 | Function: VerifyFiles(FILE *ptr_in, char *sourcePath,
 |                                     char *destPath, int required)
 | Author  : william h robertson
 |
 | Purpose : Verify the required files are all present
 |
 | Params  : ptf:    File pointer to read input data
 |           sourcePath: Name of the source path
 |           destPath  : Destination path where files are copied to
 |           required  : 1 means same file is required in both dirs
 |                       0 means only source file has to exist
 | Returns : nFiles - The number of files it found
 |
 | Usage   : nFiles = VerifyFiles(ptr_in, source, dest, required);
 |
 | Requires: <sys/stat.h>
 |           <error.h>
 |
 | Notes   : Function limited to 32-bit filesize (around 4GB)
 |
 | --Ver--  ---Date---  ----By----  ---Description of the Change---
 |     0    07-20-2016      whr     Initial release
 |
 *------------------------------------------------------------------*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define BUFF 1024

int  doesFileExist(const char* filename);

int
VerifyFiles(FILE *ptr_in, char *sourcePath, char *destPath, int required)
{
   char line[BUFF];
   char sourceFile[BUFF];
   char destFile[BUFF];
   int  nFiles  = 0;
   int  nErrors = 0;
   unsigned len_line = 0;
   long filePos = ftell(ptr_in);
```

```c

printf("Source of Files: %s\n", sourceFile);
printf("Dest   of Files: %s\n", destFile);
printf("Press [Enter] to continue");
getchar();

   while (fgets(line, BUFF, ptr_in) != NULL)
   {
      printf("Line just read %s", line);
      len_line = (unsigned) strlen(line);
      if (len_line > 1)
         line[len_line - 1] = '\0'; // Remove \n
      sourceFile[0] = destFile[0] = '\0';
      strncat(sourceFile, sourcePath, BUFF);
      strncat(sourceFile, line  , BUFF);

      fgets(line, BUFF, ptr_in);
      len_line = (unsigned) strlen(line);
      if (len_line > 1)
         line[len_line - 1] = '\0'; // Remove \n
      strncat(destFile, destPath, BUFF);
      strncat(destFile, line  , BUFF);

printf("Checking file ok\n");
      if(doesFileExist(sourceFile))
      {
         ++nFiles;               // Only increment nFiles once
printf("file ok nFiles = %d\n", nFiles);
      }
      else
      {
         ++nErrors;
         printf(" Source File: %s Missing!\n", sourceFile);
      }

      if(required && !doesFileExist(destFile))
      {
         ++nErrors;
         printf(" Destination File %s Missing\n", destFile);
      }
   }

   if(nErrors)
```

```c
    {
        printf("\nThere were %d Errors, EXITING program -- Files not copied\n", nErrors);
        printf("Press [Enter] key to return to your OS");
        getchar();
        exit(EXIT_FAILURE);
    }
    else
    {
        printf("\nThere were NO Errors. It should copy %d files.\n", nFiles);
    }

    fseek(ptr_in, filePos, SEEK_SET);    // Restore file position

    return nFiles;
}

```