```
/**
* Syed Shah
* 213927942
* October 6, 2017
* extractFile.c - This is my solution to the third lab of eecs2031.
* For this part of the lab I used the dumpsector function to output the
* characters line by line for all the files. I created a new file on my disk
* which would be named the same thing as the input file name from the user.
**/

#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <string.h>
#include <ctype.h>
#include "dos2sd.h"

static void dumpSector(FILE *fd, struct ATRSSDISK *disk, int id)
{
        int i, j;
        for(j=0;j<ATR_SECTOR_SIZE;j+=32) {
                for(i=0;i<32;i++) {
                        if(isprint(disk->sector[id][i+j]))
                                fprintf(fd,"%c", disk->sector[id][i+j]);
                        if(disk->sector[id][i+j] == '\n')
                                fprintf(fd,"\n");
                }
        }
}


static void extractFile(FILE *fd, struct ATRSSDISK *disk, char *input)
{
        int sector, entry, i, count, start, baseFileNumber, exist = 1;
        char name[9], ext[4], inN[9], inE[4];

        FILE *fp = NULL;
        baseFileNumber = 0;
        for(sector=361;sector<=368;sector++) {
                for(entry=0;entry<ATR_SECTOR_SIZE;entry+=16) {
                        if(disk->sector[sector-1][entry] == 0x042) {
                                for(i=0;i<8;i++)
                                        name[i] = disk->sector[sector-1][entry+5+i];
                                name[8] = '\0';
                                for(i=0;i<3;i++)
                                        ext[i] = disk->sector[sector-1][entry+13+i];
                                ext[3] = '\0';
                                count = disk->sector[sector-1][entry+1]|disk->sector[se
ctor-1][entry+2]<<8;
                                start = disk->sector[sector-1][entry+3]|disk->sector[se
ctor-1][entry+4]<<8;
                                sscanf(input, "%[^.]  .%s", inN, inE);
                                if(!strcmp(name, inN) && !strcmp(ext, inE)){
                                        exist = 0;
                                        sprintf(input, "%s.%s", name, ext);
                                        fp = fopen(input, "w");
                                        for(i = start-1; i < (start+count -1); i++){
                                                dumpSector(fp, disk, i);
                                        }
                                }
                        }
                        baseFileNumber++;
                }
```

```
        }
        if(exist)
                printf("The file %s.%s does not exist\n", inN, inE);
}



int main(int argc, char *argv[])
{

        struct ATRSSDISK *disk;

        if(argc != 3) {
                fprintf(stderr,"usage: %s disk\n", argv[0]);
                exit(1);
        }
        if((disk = readDisk(argv[1])) == (struct ATRSSDISK *)NULL) {
                fprintf(stderr,"Unable to read disk %s\n", argv[1]);
                exit(1);
        }
        extractFile(stdout, disk, argv[2]);
        freeDisk(disk);
        return 0;
}
```