

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include "dos2sd.h"
```

```
/*WAS MADE BY YUSUF ADISAPUTRO 213533088*/
```

```
static void listFiles(FILE *fd, struct ATRSSDISK *disk, char input[])
{
    int sector, entry, i, count, start, baseFileNumber;
    char name[9], ext[4];

    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[sector-1][entry+2]<<8;
                start = disk->sector[sector-1][entry+3]|disk->sector[sector-1][entry+4]<<8;

                int check = checkingFile(name, input);
                int check2 = checkingFileExt(ext, input);
                if(check == 1 && check2 == 1){
                    int fileSize = exploringTheData(disk, start, count);
                    toPrint(name, ext, start, count, fileSize);
                    break;
                }else{
                    printf("Checking..\n");
                }
            }
        }
        baseFileNumber++;
    }
}
```

```
int exploringTheData(struct ATRSSDISK *disk, int head, int size){
    int i, fileSize, biggest;
    fileSize = 0;
    biggest = head + size;

    for (i = head-1; i < biggest-1; i++){
        fileSize = fileSize + disk->sector[i][127];
    }
    return fileSize;
}
```

```
void toPrint(char *name, char *ext, int start, int count, int size){

    int list[count];
    int i;
    for (i = 0; i < count; i++){
        list[i] = start;
        start = start + 1;
    }
}
```

```

    printf("%s.%s sector list ", name, ext);
    for (i = 0; i < count; i++){
        printf("%d ", list[i]);
    }

    printf(" Total file size %d\n", size);

}

int checkingFile(char name[], char input[]){
    int i, count;
    count = 0;
    for(i = 0; input[i] != '\0'; i++){
        if(input[i] == name[i]){
            count++;
        }
    }
    if(count > 2){
        return 1;
    }else{
        return 0;
    }
}

int checkingFileExt(char ext[], char input[]){

    int i, finger, count;
    count = 0;
    finger = 0;
    for(i = 0; input[i] != '\0'; i++){
        if(input[i] == ext[finger]){
            count++;
            finger++;
        }
    }
    if(count > 0){
        return 1;
    }else{
        return 0;
    }
}

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);
    }
    listFiles(stdout, disk, argv[2]); /* put it in atari offset notation 1..720 */
    freeDisk(disk);
    return 0;
}

```