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

/*
FAHAD AHMED KHAN
214468888
5TH OCT 2017
THE CODE EXTRACTS FILES FROM THE DISK AND SAVES THEM IN LOCAL DISK DRIVE.
 */

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

  FILE *f;


  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;

        char na[9];
        char ex[4];

        sscanf(fileName, "%[^.]%*[.]%s", na,ex);   /* getting the first part of the nam
e to compare in the next step*/
        if(strcmp(na, name) == 0){
          f = fopen(fileName, "w");           /* creating a new file with the name of t
h argument of the command line */
                        break;
                }
      }
      baseFileNumber++;
    }
  }

  if (f == NULL)
  {
    printf("Error opening file!\n");
    exit(1);
  }

  int sec = start-1;    /* for offset notation of the sector numbers */
  int fp = 0; /* forward pointer */
  int bit = 0b1111111111;  /* for getting rid of the file # part of byte 125 */
  int h = 0;

  while(1){
    fp = (disk->sector[sec][126] | disk->sector[sec][125]<<8) & bit;          /* makin
g the 10 bit forward pointer*/
```

```
                for(entry = 0; entry < 125; entry++){
                       c = disk->sector[sec][entry];
                       fprintf(f, "%c", c);
                }


                if(fp == 0){                                /* forward pointer becomes zero
 on the last sector in the file */
                        break;
                }
                sec =fp - 1;
  }
  fprintf(f, "\n");


  fclose(f);
}



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]); /* put it in atari offset notation 1..720 */
  freeDisk(disk);
  return 0;

}
```