

```

10 REM Amstrad CPC File Recovery Program
20 REM      by Mark Cummins
30 REM
40 MEMORY &7FFF:CLS
50 'Poke in machine code
60 address=&B000
70 FOR i=1 TO 23
80 sum=0
90 FOR j=1 TO 28
100 READ code$:byte=VAL("&"+code$)
110 POKE address,byte:address=address+1:sum=sum+byte
120 NEXT j
130 READ check:IF check<>sum THEN PRINT "Error in data line";990+i*10:wrong=1
140 NEXT i:IF wrong=1 THEN END
150 PRINT "Data loaded correctly.":PRINT "RSX command 'IRECOVER' initialized.":C
ALL &B000:END
1000 DATA 01,7C,01,21,09,01,CD,D1,BC,C9,3E,01,CD,55,01,3E,FF,DF,0E,01, 2777
1010 DATA 3E,09,CD,55,01,3E,03,DF,0E,01,3E,04,CD,55,01,DD,21,91,01,21, 2351
1020 DATA C1,01,1E,00,16,00,DD,7E,00,FE,00,CA,40,01,4F,DF,0E,01,30,04, 2003
1030 DATA DD,23,10,E7,DD,7E,00,32,96,01,FE,41,20,00,FE,C1,20,00,FE,01, 2307
1040 DATA 20,0E,3E,02,32,95,01,10,0C,3E,00,32,95,01,10,05,3E,01,32,95, 1163
1050 DATA 01,21,C1,01,1E,00,3A,95,01,57,3A,96,01,4F,06,01,DF,0E,01,D2, 2064
1060 DATA 40,01,0C,C5,01,FF,01,ED,4A,C1,04,3E,05,00,20,02,10,EA,21,C1, 1944
1070 DATA 01,23,7E,FE,E5,CA,17,01,20,7E,FE,E5,CC,A2,00,01,1F,00,ED,4A, 2616
1080 DATA 10,EB,23,E5,CD,B5,00,E1,FE,59,CC,11,01,DD,21,BC,01,CD,6F,01, 2971
1090 DATA C9,0E,01,7E,CD,5A,00,0C,23,79,FE,09,20,F5,3E,2E,CD,5A,00,7E, 2240
1100 DATA CD,5A,00,0C,23,79,FE,0C,20,F5,DD,21,97,01,CD,6F,01,7E,FE,00, 2552
1110 DATA 20,07,DD,21,AB,01,CD,6F,01,CD,09,00,30,FB,CD,01,00,CD,06,00, 2673
1120 DATA CD,04,00,FE,59,20,0E,FE,79,20,0A,FE,4E,20,0C,FE,6E,20,00,10, 2166
1130 DATA E5,3E,59,CD,5A,00,C9,3E,4E,CD,5A,00,C9,20,3E,00,77,23,C9,3E, 2400
1140 DATA 05,CD,55,01,21,C1,01,1E,00,3A,95,01,57,3A,96,01,4F,06,01,DF, 2006
1150 DATA 0E,01,30,10,0C,C5,01,FF,01,ED,4A,C1,04,3E,05,00,20,02,10,EB, 1861
1160 DATA 3E,01,CD,55,01,3E,00,DF,0E,01,3E,09,CD,55,01,3E,10,DF,0E,01, 2356
1170 DATA C9,32,0D,01,E1,22,BF,01,21,0D,01,CD,04,BC,D0,22,0E,01,79,32, 2692
1180 DATA 90,01,2A,BF,01,E5,C9,DD,7E,00,FE,00,C0,CD,5A,00,DD,23,10,F3, 2071
1190 DATA 01,01,C3,0A,00,52,45,43,4F,56,45,D2,00,00,00,00,00,00,00,00, 1253
1200 DATA 00,41,C1,01,00,00,00,20,20,20,52,65,63,6F,76,65,72,3F,20,20, 1216
1210 DATA 59,2F,4E,29,20,3A,00,20,20,20,20,65,70,74,29,00,00,00,00,00, 099
1220 DATA 00,00,00,00,0D,0A,00,00,00,00,00,00,00,00,00,00,00,00,00,00, 47

```

;RECOVER (c) 1986 by Mark Cummins

;  
; This routine allows the user to selectively unerase any disk files that have  
; been erased. If a file is longer than 16K then it will be displayed twice, the  
; second time with the word (ext) after the filename. To recover the whole file,  
; press "Y" each time the filename is displayed. Some files may have been  
; overwritten and still have their filename in the directory. In this case the  
; file probably won't load.  
;-----

    ; set up the RSX table  
    ld bc,table  
    ld hl,rsxwork  
    call &bcd1  
    ret

;  
; set disc messages off  
RECOVER ld a,129  
        call bios  
  
        ld a,&ff  
        rst 3  
        defw cmd\_far\_addr

;  
; set retry count to a smaller value  
        ld a,137  
        call bios  
  
        ld a,3  
        rst 3  
        defw cmd\_far\_addr

;  
; read sector 0 with different sector values until right format is found  
        ld a,132  
        call bios  
        ld ix,sect\_value

loop   ld hl,buffer  
        ld e,&00  
        ld d,0  
        ld a,(ix)

```

cp 0 ;If A = 0 then the format can't be determined
jp z, seton ;so set messages and retry count to default
;and return to Basic.

ld c, a ;C contains first sector number of the format.
rst 3 ;Do a FAR CALL to BIOS 'read sector' routine
defw cmd_far_addr ;at this address.
jr c, direct ;if the format tested is correct then jump,
inc ix ;else IX holds address of next format to test
jr loop ;and we go round again.

;set up the directory track & sector values according to the disk's format.
direct ld a, (ix) ;Load A with the first sector number
ld (sector), a ;and store it.
cp &41
jr z, system ;Jump to System if first sector number = &41.
cp &c1
jr z, data ;Jump to Data if first sector number = &c1.
cp &01
jr z, ibm ;Jump to IBM if first sector number = &01.
system ld a, 2 ;Directory on track 2 if System format.
ld (track), a
jr start
data ld a, 0 ;Directory on track 0 if Data format.
ld (track), a
jr start
ibm ld a, 1 ;Directory on track 1 if IBM format.
ld (track), a

;read in all four sectors that make up the directory
start ld hl, buffer ;HL holds address of directory buffer
ld e, &00 ;Drive = A.
ld a, (track)
ld d, a ;D holds the directory track number.
ld a, (sector)
ld c, a ;C holds first sector number of directory.
ld b, 1 ;Set 'no. of sectors loaded' counter to 1.
rst 3 ;Do a FAR CALL to BIOS 'read sector' routine
defw cmd_far_addr ;at this address.
jp nc, seton ;If unsuccessful then set messages and retry
;count to default and return to Basic.

inc c ;Inc. C to point at next directory sector.
push bc
ld bc, 511
adc hl, bc ;HL holds address of directory buffer for the
pop bc ;next sector.
inc b ;Add 1 to 'no. of sectors loaded' counter.
ld a, 5
cp b ;If all four directory sectors have been
jr z, query ;loaded then jump,
jr sl ;else go round and read in next sector.

query ld hl, buffer ;HL points to address of the directory buffer.
ql inc hl
ld a, (hl) ;A holds first byte of the filename.
cp &e5 ;If &E5 then it is the end of the directory so
jp z, repair ;rewrite the directory sectors and finish.
dec hl
ld a, (hl) ;A holds the file active/erased byte.
cp &e5 ;If &E5 then the file has been erased so jump.
call z, erased
ld bc, 31
adc hl, bc ;HL holds address of next filename entry.
jr ql ;Go round again.

erased inc hl ;HL is on first byte of filename.
push hl ;Print the filename and ask whether to recover
call fileprint
pop hl
cp "Y" ;If A = "Y" then the user wants to recover the
call z, unerase ;file so jump to the unerase routine.
ld ix, 14
call sprint ;Print a linefeed after the filename.
ret

fileprint ld c, 1 ;Print the first 8 letters of the filename.
fl ld a, (hl)
call &bb5a
inc c
inc hl
ld a, c
cp 8
jr nz, fl
ld a, "." ;Then print the filename extension divider

```

```

call &bb5a
f2 ld a,(hl) ;and the 3 letter filename extension.
call &bb5a
inc c
inc hl
ld a,c
cp 12
jr nz,f2
ld ix,recov ;Display a message asking the user whether or
call sprint ;not to recover the file.
ld a,(hl) ;A holds entry extent number.
cp 0
jr z,flush
ld ix,ext ;If entry extent number > 0 then tell the user
call sprint ;this is second.directory entry for the file.
flush call &bb09 ;Flush the keyboard buffer.
jr c,flush
f3 call &bb01 ;Switch the text cursor on.
call &bb06 ;Wait for a character from the keyboard.
call &bb04 ;Switch off the text cursor.
cp "Y" ;Jump to routine f4 if answer is "y" or "Y".
jr z,f4
cp "y"
jr z,f4
cp "N" ;Jump to routine f5 if answer is "n" or "N".
jr z,f5
cp "n"
jr z,f5
jr f3 ;The wrong key was pressed so wait for another
f4 ld a,"Y" ;Print "Y" on the screen.
call &bb5a
ret
f5 ld a,"N" ;Print "N" on the screen.
call &bb5a
ret

unerase dec hl ;HL points to the file's active/erased byte.
ld a,0 ;Set the byte to 0 (active).
ld (hl),a ;HL points to first byte of filename.
inc hl
ret

;write back all four sectors that make up the directory
repair ld a,133 ;A holds BIOS 'write sector' command name.
call bios ;Find and store the address of the command.

ld hl,buffer ;HL holds address of directory buffer.
ld e,&00 ;Drive = A.
ld a,(track)
ld d,a ;D holds the directory track number.
ld a,(sector)
ld c,a ;C holds first sector number of directory.
ld b,1 ;Set 'no. of sectors written' counter to 1.
r1 rst 3 ;Do a FAR CALL to BIOS 'write sector' routine
defw cmd_far_addr ;at this address.
jr nc,seton ;If write unsuccessful then set messages and
;retry count to default and return to Basic.
inc c ;Inc. C to point at next directory sector.
push bc
ld bc,511 ;HL holds address of directory buffer for the
adc hl,bc ;next sector.
pop bc
inc b ;Add 1 to 'no. of sectors written' counter.
ld a,5
cp b ;If all 4 directory sectors have been written
jr z,seton ;then jump,
jr r1 ;else go round again.

;set disk messages on again
seton ld a,129 ;A holds BIOS 'set message' command name.
call bios ;Find and store the address of the command.

ld a,&00 ;&00 = disk messages enabled
rst 3 ;Do a FAR CALL to BIOS 'set message' routine
defw cmd_far_addr ;at this address.

;set retry count to default value
ld a,137 ;A holds BIOS 'set retry count' command name.
call bios ;Find and store the address of the command.

ld a,16 ;A holds retry count (16 is the default).
rst 3 ;Do FAR CALL to BIOS 'set retry count' routine
defw cmd_far_addr ;at this address.
ret ;Return to Basic.

```

```

bios      ld (cmd_name),a          ;A holds the BIOS command name.
          pop hl
          ld (basic),hl          ;Store return pointer.
          ld hl,cmd_name         ;HL points to the command name address.
          call &bcd4            ;Ask the kernel where it is.
          ret nc                ;Return to Basic if it is not found.

          ld (cmd_far_addr+0),hl ;Store the routine's address
          ld a,c                ;and ROM number.
          ld (cmd_far_addr+2),a
          ld hl,(basic)
          push hl               ;Restore the return pointer.
          ret

sprint    ld a,(ix)             ;Print a string routine.
          cp 0
          ret z
          call &bb5a
          inc ix
          jr sprint

table     defw names            ;RSX table.
          jp RECOVER

names     defb "RECOVER", "R"+128
          defb 0

rsxwork   defs 4

cmd_name  defb 0               ;Storage area for BIOS command name.
cmd_far_addr  defs 3          ;Area for storing BIOS address and ROM no.
sect_value  defb 541,&ci,501,&00
                                     ;System,Data,IBM sector values

track     defb 0
sector    defb 0
recov     defb " Recover? (Y/N) :",0
ext       defb " (ext)",0,0,0,0,0,0,0,0,0
lf        defb 13,10,0
basic     defw 0000
buffer    defs 2048

```