

```

;-----
;      Amstrad ROM
;      By R.A.Waddilove
;-----
WRITE "CWTAROM.BIN"
NOLIST
LIMIT &FFFF
ORG &C000
;-----
;  Reserves 32 bytes for workspace:
;  0-17 used by DOUBLE
;-----
;      ROM Prefix
;-----
DEFB 1      ;ROM type...background
DEFB 1      ;mark 1
DEFB 1      ;version 1
DEFB 1      ;modification 1
;-----
;      Command table
;-----
DEFW name_table
JP initialise_rom      ;power-up/
reset entry
JP help_cwta          ;help screen
JP help_cwta
JP double              ;double height
print
JP double
;-----
;      Name table
;-----
.name_table
DEFB "CWTA RO", "M"+&80      ;bad name
-> can't power-up with command
DEFB "HEL", "P"+&80          ;HELP
DEFB "CWTAHEL", "P"+&80      ;CWTA
prefix
DEFB "DOUBL", "E"+&80        ;DOUBLE, @a$
;DOUBLE, @a$
DEFB "CWTADOUBL", "E"+&80    ;CWTA
prefix
DEFB 0
;-----
;      Power-up/Reset intialisation
;      AF,BC corrupted
;-----
.initialise_rom
PUSH DE:PUSH HL      ;save
DE/HL
CALL print_string      ;print
power-up message
DEFB 13,10
DEFB "Computing with the Amstrad ROM"
DEFB 13,10,10
DEFB 0
POP HL:POP DE          ;restore
HL/DE
AND A
LD BC,32:SBC HL,BC      ;grab 32
bytes from top of memory
SCF
RET
;-----
;      General string print subroutine
;      AF,HL corrupted
;-----
.print_string
POP HL      ;get string
address
.sp_loop
LD A,(HL):CALL &8B5A      ;print cha-
racter
INC HL
OR A          ;done?

```

```

JR NZ,sp_loop
JP (HL)
;-----
;      HELP
;      Print syntax and function of commands
;      AF,HL corrupted
;-----
.help_cwta
CALL print_string
DEFB "Computing with the Amstrad
ROM",1fl,13
DEFB "HELP" - list all com-
mands",1fl,13
DEFB "DOUBLE,@a$ - double height pri-
nt",1fl,13
DEFB "Names can be prefixed with
CWTA",1fl,13
DEFB 7,0
RET
;-----
;      DOUBLE
;      Double height print
;      AF,BC,DE,HL corrupted
;-----
.double
DEC A:JP NZ,help_cwta      ;one
parameter?
LD L,(IX+0):LD H,(IX+1)      ;HL
points to string descriptor
LD A,(HL)
;A=length
AND A:RET 2
;abort if 0
LD (IY+17),A
;workspace+17=length
INC HL:LD E,(HL):INC HL:LD D,(HL)
;DE=address
.loop1
PUSH DE
LD A,(DE)      ;get
character
CALL &8BA5
;HL=data address
CALL &8B06:PUSH AF
;enable lower ROM
PUSH IY:POP DE      ;DE=-
workspace
LD B,8      ;copy
data to workspace
.loop2
LD A,(HL):LD (DE),A
INC DE:LD (DE),A
INC DE:INC HL
DJNZ loop2
POP AF:CALL &8B0C      ;res-
tore lower ROM state
LD A,255
;define top half of character
PUSH IY:POP HL
CALL &8BA8
LD A,254
;define bottom half of character
PUSH IY:POP HL
LD DE,8:ADD HL,DE
CALL &8BA8
CALL print_string
;print big character
DEFB 255,8,10,254,11,0
POP DE:INC DE      ;next
character
DEC (IY+17):JR NZ,loop1      ;all
done?
RET
list
END
**** END *

```