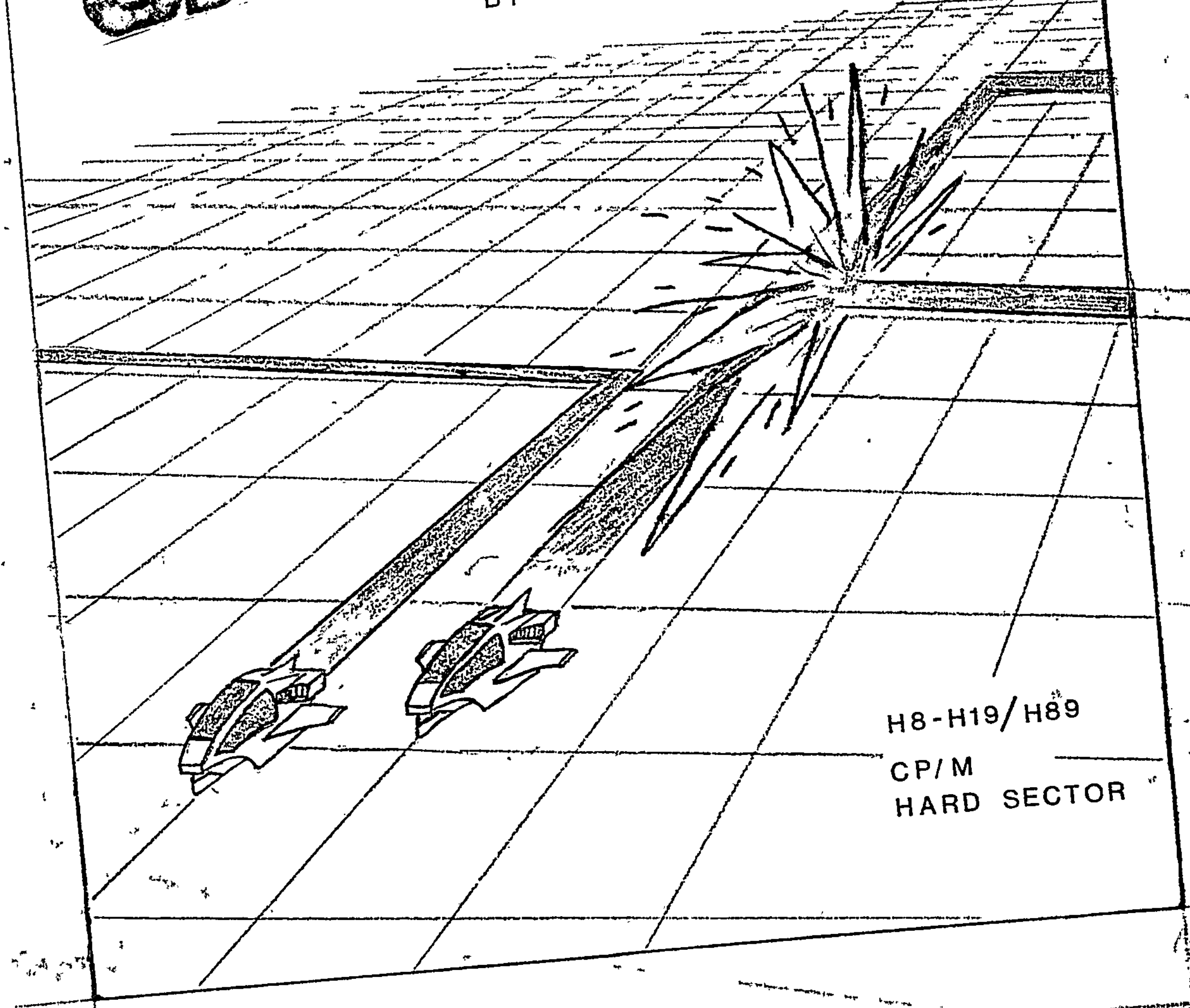
others are heavily industrialized ... why it is ... mans and other inform ... of the

COMPTEC SOFTWARE

PRESENTS

CYCLEMANIA

BY LES BIRD



H8-H19/H89

CP/M
HARD SECTOR

Program Control

PC/M

Cycle
MASTER CONTROL CYCLE#

J
E

MCP

Evil CP/M

The evil lord Cycle has Taken over The electronics kingdom. Cycle mania was once a peacful past Time. now its certain death. Your mission is survive The unlimited wave of cycle. The Action is fast & you have to be on your guard. You start up, your cycle is at full power, Quickly you take a sharp left bank then a right, watch out! These a force field in front of you That you opponet left. Suddenly all The wall fall, You've survied for now but wait Till next Time

Year ~~2976~~ ~~2215~~ ~~2743~~ 28th Century

Description:

As you ~~push~~ speed through the GAME GRID you will encounter up to three ROBOT CYCLES. These cycles have bee programmed to kill by the CPU. Your mission is to survive by avoiding the anti-matter light traces. The CPU will stop at nothing to finish you off.

2-23-83 By

It the year of 2300. Baseball & football are absolete. There are no more parks with green grass & trees. Everything is machine now. Now the major sport is Suicide. Crimanals are taken from prison & put in their machines to duel against a computer cyclone. If they loose their sentence is justifyed. If they win they are let free. You control the computer you must help eliminate the overcrowding of prisons.

The Roman empire as risen again. ~~Thee~~ Their great colusium rebuilt with steel. This time two men are put up against eachother in the Lion machines. You are one of the men ~~to~~ chosen to compete. Live or die. The controls are left up to you.

It is the year 2108. The U.S.S.R. is just about to attack the U.S. Only you can save the U.S.A. You have been given a special message that must be delivered to the top brass in Washington. If you make it you will save the U.S.A. if not it is total war. Your mission will not be easy. You have been given a light cycle to travel but the U.S.S.R. does sent a "destroyer drone" out to destroy you & your machine. Escape from it & save the U.S.A.

ME A8/19/89 Heath~~kit~~ H-8 and H19/H89

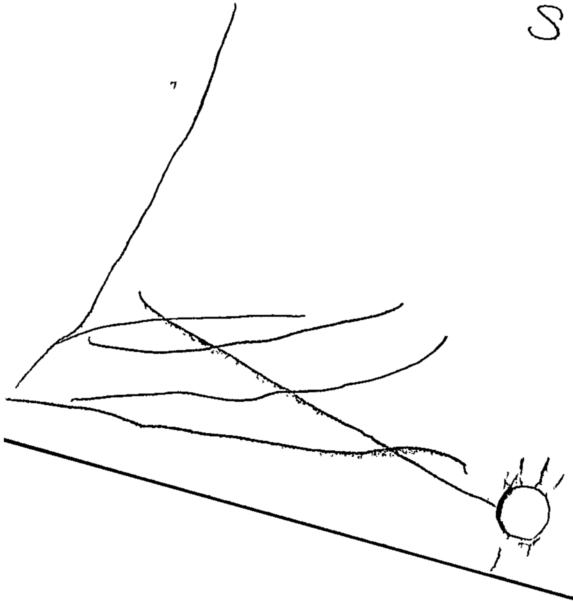
CP/M
HARD SECTOR

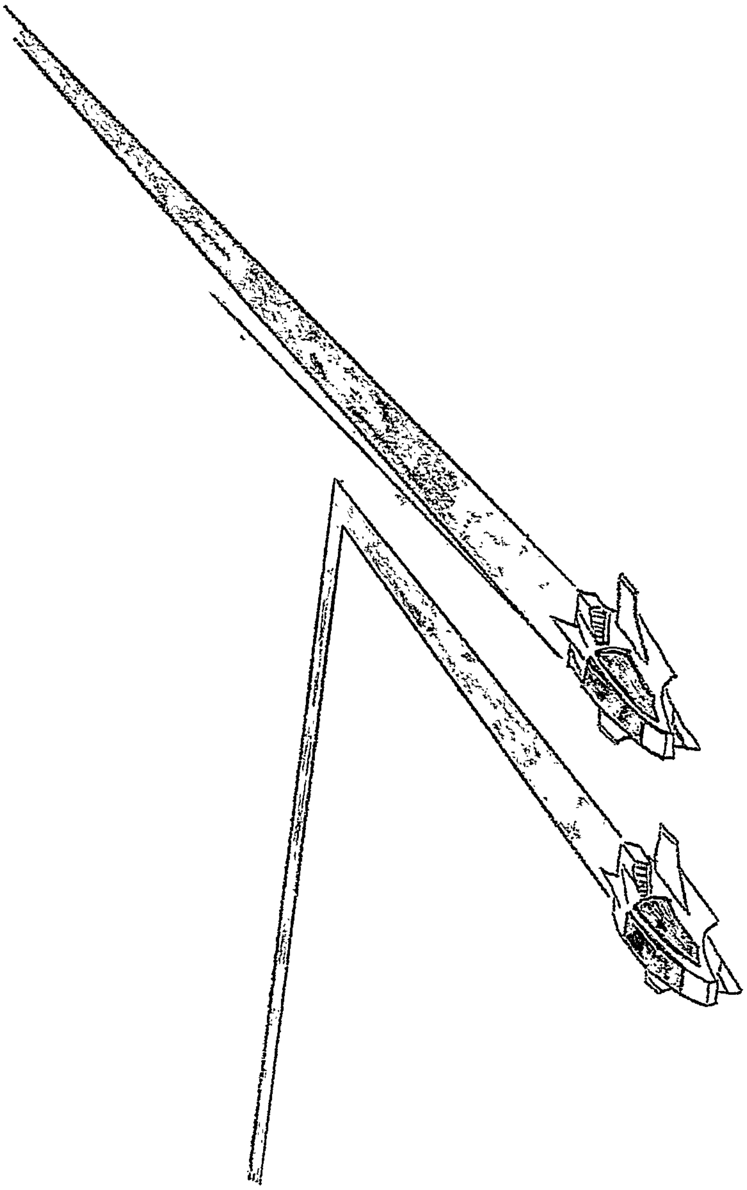
COMPTEC SOFTWARE presents

SOFTWARE SOFTWARE

CYCLEMANIA

1 2 3 4 5 6 7 8 9 10





```
;       ***********************************************
;       *               H19 LIGHT CYCLE               *
;       *                                             *
;       *        LAST UPDATE:   July 9, 1983          *
;       *         PROGRAMMER:   Les Bird              *
;       *           LANGUAGE:   8080 Assembly Language*
;       *            COMPANY:   CompTec software      *
;       *   OPERATING SYSTEM:   CP/M ver 2.2          *
;       ***********************************************
;
;       JUMP VECTORS            04/10/83
;
tpa     equ     0ec4h
;
cr      equ     0dh
lf      equ     0ah
bs      equ     08h
ap      equ     27h
esc     equ     1bh
hf1     equ     'S'
hf2     equ     'T'
hf3     equ     'U'
hf4     equ     'V'
hf5     equ     'W'
hf7     equ     'P'
hf8     equ     'Q'
hf9     equ     'R'
hram    equ     'z'     ; Terminal Reset
bdos    equ     0005h
fcb     equ     005ch
dma     equ     0080h
clock   equ     000bh
;
;       BDOS OPERATIONS
;
conin   equ     01h
conout  equ     02h
direct  equ     06h
dinput  equ     0ffh
pstring equ     09h
rbuff   equ     0ah
constat equ     0bh
seldsk  equ     0eh
openf   equ     0fh
closef  equ     10h
delf    equ     13h
readf   equ     14h
writef  equ     15h
makef   equ     16h
curdsk  equ     19h
dmaset  equ     1ah
;
        org     tpa
;
        jmp     start   ; start program at BEGIN
;
cls:    jmp     10dh
zsmem:  jmp     110h
ceol:   jmp     113h
crlf:   jmp     116h
```

```
deletec:jmp     122h
home:   jmp     143h
process:jmp     146h
hset:   jmp     149h
clr25:  jmp     14ch
hrset:  jmp     14fh
output: jmp     152h
input:  jmp     155h
wait:   jmp     158h
bentry: jmp     15eh
savall: jmp     161h
retall: jmp     164h
adj:    jmp     167h
update: jmp     16ah
rsmem:  jmp     16dh
ssmem:  jmp     170h
graphix:jmp     173h
xgraphix:
        jmp     176h
reverse:jmp     179h
xreverse:
        jmp     17ch
cursoff:jmp     17fh
curson: jmp     182h
cursl:  jmp     185h
cursb:  jmp     188h
spmsg:  jmp     19ah
show:   jmp     19dh
delay:  jmp     1a0h
;
; STORE HIGH SCORES ON DISK AND REBOOT SYSTEM
;
boot:
restore:        ; Reboot system
        mvi     a,hram
        call    process
        call    savall  ;
        call    wrhigh  ; write highscores on disk
        call    closefile
        call    retall  ;
        lhld    oldstack; get returning address for ccp
        sphl            ; set stack pointer to new stack
        ret             ; back to ccp
;
;       INPUT A LINE FROM THE KEYBOARD
;
inbuf:  push    h
        call    wait
        pop     h
        cpi     cr
        rz
        cpi     bs
        jz      inbuf2
        cpi     3       ; ctrl-c
        jz      xinbuf
        cpi     20h
        jc      inbuf
        cpi     61h
        jc      inbuf1
        ani     5fh
```

```
inbuf1: mov     m,a
        push    h
        call    output
        pop     h
        inx     h
        jmp     inbuf
inbuf2: dcx     h
        mov     a,m
        cpi     1
        jz      inbuf3
        push    h
        mvi     a,bs
        call    output
        mvi     a,'_'
        call    output
        mvi     a,bs
        call    output
        pop     h
        jmp     inbuf
inbuf3: mvi     a,7     ; bell
        push    h
        call    output
        pop     h
        inx     h
        jmp     inbuf   ; loop
xinbuf: pop     b
        jmp     begin
; sets up File Control Block for high scores
fcbset: lda     drive
        sta     fcb     ; drive #
        lxi     h,filename
        lxi     d,fcb+1
fcbset1:mov     a,m
        ora     a
        jz      fcbset2 ; write .TYP
        stax    d
        inx     d
        inx     h
        jmp     fcbset1
fcbset2:lxi     h,filetype
fcbset3:mov     a,m
        ora     a
        jz      fcbset4
        stax    d
        inx     h
        inx     d
        jmp     fcbset3
fcbset4:lda     extnum
        sta     fcb+12
        lda     reccnt
        sta     fcb+15
        lda     currec
        sta     fcb+32
        lhld    recnum
        shld    fcb+33
        xra     a
        sta     fcb+35
        ret
; open the HIGHSCORE file on the disk
openfile:
        call    fcbset
        mvi     c,openf
        lxi     d,fcb
        call    bdos
        cpi     0ffh    ; no file to open
        rnz
makefile:
        call    fcbset
        mvi     c,makef
        lxi     d,fcb
        call    bdos
        cpi     0ffh
        jz      nserror ; no space
        ret
; close file
closefile:
        mvi     c,closef
        lxi     d,fcb
        call    bdos
        cpi     0ffh
        jz      nferror
        ret
; read data from file
readfile:
        mvi     c,readf
        lxi     d,fcb
        call    bdos
        ret
; write data to file. 0ffh=end of file
writefile:
        mvi     c,writef
        lxi     d,fcb
        call    bdos
        ret
; delete file
delfile:mvi     c,delf
        lxi     d,fcb
        call    bdos
        ret
;
; USER PROGRAM STARTS HERE
;
start:          ; Logical Start of Program
        call    highfile        ; Read or Create High Score on disk
        lxi     h,0700h
        shld    time
;
count1: ds      2       ; 32 counts
begin:  lxi     h,0020h
        shld    count1
;                       ; cls patch
        mvi     a,20h   ; space
        sta     1d0h    ; patch clear screen routine
;
        lxi     h,intro
        call    show    ; print heading
        call    cursoff
        mvi     a,1
        sta     lev2
begin1: call    diskill ; display skill level
```

```
        call    copyright
        lhld    count1
        dcx     h
        mov     a,h
        ora     l
        jz      cycle19 ;
        shld    count1
        mvi     a,'}'
        sta     boxchar
        call    box1
        call    begin3
        mvi     a,'|'
        sta     boxchar
        call    box7
        call    begin3
        call    copyright
        call    box1
        call    begin3
        mvi     a,'i'
        sta     boxchar
        call    box7
        call    begin3
        jmp     begin1
begin3: call    input
        ora     a
        jnz     begin2
        ret
begin2: pop     b       ; return address
        cpi     'a'     ; lower-case character?
        jc      b2      ; no, then skip
        ani     5fh     ; else convert to upper-case
b2:                     ;
        lxi     h,0020h
        shld    count1
        cpi     esc
        jz      escvec
        cpi     'Q'
        jz      boot
        cpi     'W'
        jz      cycle19
        cpi     '1'     ; same as F1
        jz      pgame   ; preset game parameters
        cpi     '2'     ; same as F2
        jz      highscore
        cpi     '3'     ; same as F3
        jz      optmenu ; display option menu
        cpi     '5'     ; same as BLUE key
        jz      restore ; quit
        jmp     begin1
escvec: call    wait
        cpi     hf1
        jz      pgame
        cpi     hf2
        jz      highscore
        cpi     hf3
        jz      optmenu ; print option menu
        cpi     hf7
        jz      boot
        jmp     begin1
;
```

```
optmenu:                ; option menu
        call    spmsg   ; print following message
        db      '[C]@',6,28,'C Y C L E   M A N I A'
        db      '@',7,28,'[G]zzzzzzzzzzzzzzzzzzzzzz[g]'
        db      '@',8,34,'OPTIONS MENU'
        db      '@',12,26,'(1) to select speed of play'
        db      '@',13,26,'(2) to select level of play'
        db      '@',14,26,'(3) to select control options'
        db      '@',16,26,'(4) to return to MAIN MENU'
        db      0
        call    wait    ; get selection
        cpi     '1'     ; was the '1' key pressed ?
        jz      selspd  ; select speed level
        cpi     '2'     ; '2' ?
        jz      sellev  ; select level
        cpi     '3'     ; '3' ?
        jz      controls; select control options
        jmp     begin   ; return to menu
;
selspd: lxi     h,mess10
        call    show
        call    wait
        cpi     '1'
        jz      fast
        cpi     '2'
        jz      slow1
        cpi     '3'
        jz      slow2
        cpi     '4'
        jz      slow3
        jmp     selspd
fast:   lxi     h,0100h
        shld    time
        mvi     a,1
        sta     skill
        sta     lev2
        jmp     optmenu
slow1:  lxi     h,0200h
        shld    time
        mvi     a,2
        sta     skill
        jmp     optmenu
slow2:  lxi     h,0700h
        shld    time
        mvi     a,3
        sta     skill
        jmp     optmenu
slow3:  lxi     h,1200h
        shld    time
        mvi     a,4
        sta     skill
        jmp     optmenu
;
sellev: call    dislevell
        lxi     h,levmes1
        call    show
sellev1:call    wait
        cpi     '9'+1   ; highest level to start
        jnc     sellev1 ; higher than nine
        cpi     '1'
        jc      sellevl ; less than zero
        sui     31h     ; make binary number
        sta     lev2    ; adjust level
        call    graphix ; enter graphics mode
        call    dislevel2
        jmp     optmenu
;
diskill:lda     skill   ; get skill level
        cpi     1       ; expert skill
        jz      skill1
        cpi     2       ; pro skill
        jz      skill2
        cpi     3       ; intermediate skill
        jz      skill3
        cpi     4       ; amatuer skill
        jz      skill4
skill1: lxi     h,expmes
        call    show
        ret
skill2: lxi     h,promes
        call    show
        ret
skill3: lxi     h,intmes
        call    show
        ret
skill4: lxi     h,amames
        call    show
        ret
;
controls:       ; control option's
        call    spmsg   ; display message
        db      '[C]@',5,28,'C Y C L E   M A N I A'
        db      '@',8,34,'OPTIONS MENU'
        db      '@',12,26,'(1) normal key sequence'
        db      '@',14,26,'(2) alternate key sequence'
        db      0
        call    wait    ; get input
        cpi     '1'     ; Normal Key mode ?
        jz      cntrls1
        mvi     a,1
        sta     cntrls
        jmp     optmenu
cntrls1:xra     a       ; zero byte
        sta     cntrls
        jmp     optmenu
;
pgame:  mvi     a,4
        sta     cycleft
        lxi     h,3030h
        shld    score1
        shld    score1+2
        mvi     a,'0'
        sta     score1+4
        mvi     a,1
        sta     lev
        sta     lev1
        sta     level
        sta     cdbit2
        sta     cdbit3
        mvi     a,0
        sta     cdbit1
        sta     lev3
        call    cursoff
        lxi     h,freebie
        shld    freepnt
;
game:   call    graphix ; enter graphics mode
        call    dislevel; display level on screen
        call    xgraphix; exit
        lxi     h,grid
        call    show
        call    init
        call    place
        call    misc
;
loop1:  call    wait
        cpi     'q'
        jz      begin
        cpi     'Q'
        jz      begin
        sta     plmove
;
game1:  call    delay
        call    player
        lda     plabit
        ora     a
        jz      game
        call    computr
        lda     combit
        ora     a
        jz      game
        call    delay
        jmp     game1
;
place:  lhld    playco
        call    adj
        lda     pldir
        call    grout
        lhld    comco
        call    adj
        lda     comdir
        call    grout
        lda     lev1
        cpi     2
        rc
        jnz     place2
;
place1: lhld    comco2
        call    adj
        lda     comdir
        call    grout
        ret
;
place2: call    place1
        lhld    comco3
        call    adj
        lda     comdir
        call    grout
        ret
;
```

```
player: lhld    placo
        shld    oldco
        call    input
        ora     a
        jz      nomove
        cpi     '5'
        jz      pspeed
        cpi     ' '
        jz      pspeed
        cpi     'p'
        jz      pause
        cpi     'P'
        jz      pause
        cpi     'q'
        jz      begin
        cpi     'Q'
        jz      begin
        sta     plmove
        lda     cntrls
        ora     a       ; Test for Normal or Modified mode.
        cnz     player1 ; Modified mode.
        lda     plmove  ; Get new move.
        cpi     '4'
        jz      pleft
        cpi     '6'
        jz      pright
        cpi     '2'
        jz      pdown
;
pup:    dcr     l
        shld    placo
        mov     a,l
        cpi     3
        jz      pldead
        mvi     a,'u'
        sta     pldir
        jmp     xplayer
;
pdown:  inr     l
        shld    placo
        mov     a,l
        cpi     23
        jz      pldead
        mvi     a,'s'
        sta     pldir
        jmp     xplayer
;
pleft:  dcr     h
        shld    placo
        mov     a,h
        cpi     1
        jz      pldead
        mvi     a,'t'
        sta     pldir
        jmp     xplayer
;
pright: inr     h
        shld    placo
        mov     a,h
        cpi     80
        jz      pldead
        mvi     a,'v'
        sta     pldir
        jmp     xplayer
;
nomove: lda     plmove
        cpi     '2'
        jz      pdown
        cpi     '4'
        jz      pleft
        cpi     '6'
        jz      pright
        jmp     pup
;
pspeed: lda     speedi
        cpi     1
        jz      pspeedi
        mvi     a,1
        sta     speedi
        jmp     player
;
pspeedi:mvi     a,2
        sta     speedi
        jmp     player
;
xplayer:lhld    playco
        call    adj
        lda     pldir
        call    grout
        call    rsmem
        cpi     ' '
        jnz     pldead
        lhld    oldco
        call    adj
        jmp     plchar
;
playerl:        ; Modified mode
        lda     plmove  ; Get old move
        cpi     '4'     ; if left
        jz      pdecr   ; decrement move
;                       ; Else increment move
        lda     pldir   ; get player's direction
        cpi     's'     ; Direction 2 ?
        jz      pinr1
        cpi     't'     ; Direction 4 ?
        jz      pinr2
        cpi     'u'     ; Direction 8 ?
        jz      pinr3
;                       ; Must be Direction 6 if nothing else
pinr0:                  ;
        mvi     a,'2'   ; New Direction
        sta     plmove  ; new move
        ret             ; return
;                       ; Direction 2
pinr1:                  ;
        mvi     a,'4'
        sta     plmove
        ret
;                       ; Direction 4
pinr2:                  ;
```

```
        mvi     a,'8'
        sta     plmove
        ret
;
pinr3:                  ; Direction 8
        mvi     a,'6'
        sta     plmove
        ret
;
pdecr:                  ; Move Left
        lda     pldir   ; Get Direction of Travel
        cpi     's'     ; Down ?
        jz      pinr3   ; move left one direction
        cpi     't'     ; Left ?
        jz      pinr0
        cpi     'u'     ; Up ?
        jz      pinr1
        jmp     pinr2   ; Must be Right.
;
plchar: mvi     a,'i'
        call    grout
        call    ssmem
        call    blbon   ; block bonus
        lda     speed1
        cpi     4
        jz      pspeed2
        cpi     2
        jz      pspeed3
        ret
;
blbon:  lxi     h,blocks
        inx     h
        inx     h
;
blbon1: mov     a,m     ; get tens unit
        inr     a
        cpi     ':'     ; test for 10
        jz      blbon2
        mov     m,a
        ret             ; ten points added
;
blbon2: mvi     m,'0'   ; ascii zero
        dcx     h
        jmp     blbon1
;
addbonus:
        lxi     h,bonusmes
        call    show
        lxi     h,score1; player 1 score
        xchg            ; put in DE
        lxi     h,blocks; bonus points
        inx     d       ; 10,000's digit
        inx     d       ; 1,000's digit
        inx     d       ; 100's digit
        inx     d       ; tens position
        inx     h
        inx     h       ; tens position
;
addbonus1:
        push    h
        push    d
        mvi     a,' '   ; delimiter
        cmp     m
        jz      addbonus4
        ldax    d
        sui     30h
        add     m
        cpi     ':'
        jc      addbonus3
;
addbonus2:
        sui     0ah     ; subtract ten
        stax    d       ; put in score
        dcx     d       ; next unit
        ldax    d       ; get number
        inr     a       ; increment count
        cpi     ':'
        jz      addbonus2
;
addbonus3:
        stax    d
        pop     d
        pop     h
        dcx     h
        dcx     d
        jmp     addbonus1
;
addbonus4:
        pop     d
        pop     h
        lxi     h,3030h ; zero block count
        shld    blocks
        shld    blocks+2
        call    scorout ; update score
        ret
;
bonusmes:
        db      '@',13,28,'BONUS  + '
;
blocks: db      '0000  POINTS ',0
;
pspeed2:mvi     a,2
        sta     speed1
        ret
;
pspeed3:mvi     a,4
        sta     speed1
        jmp     player
;
grout:  push    psw
        call    graphix
        pop     psw
        call    output
        push    psw
        call    xgraphix
        pop     psw
        ret
;
pldead: mvi     a,0
        sta     plabit
        lda     cycleft
        dcr     a
        jz      gameov
        sta     cycleft
        mvi     a,1
        sta     speed1
        lxi     h,pdmes
        call    show
;
reinit: lda     lev
        cpi     1
        jz      rein1
        cpi     2
        jz      rein2
;
rein3:  mvi     a,0
        sta     cdbit3
        sta     cdbit2
        sta     cdbit1
        jmp     loop2
;
rein2:  mvi     a,0
        sta     cdbit2
        sta     cdbit1
        mvi     a,1
        sta     cdbit3
        jmp     loop2
;
rein1:  mvi     a,0
        sta     cdbit1
        mvi     a,1
        sta     cdbit2
        sta     cdbit3
;
loop2:  call    addbonus
loop2b: call    wait
        cpi     cr
        jnz     loop2b
        call    clr25
        ret
;
gameov: call    newhigh
        lxi     h,gomes
        call    show
        call    wait
        ori     20h     ; make lower case
        cpi     'y'
        jz      pgame
        cpi     'n'
        jz      begin
        cpi     'q'
        jz      restore
        jmp     gameov
;
gomes:  db      '[C]@',12,34,'[R] GAME OVER [r]@',15,20
        db      'Press [R] Y [r] to PLAY AGAIN,  [R] N [r] to QUIT.',0
;
cload:  lda     level
        cpi     1
        jz      load1
        cpi     2
        jz      load2
;
load3:  lda     cdbit3
        ora     a
        jnz     xload1
        lhld    comco3
        mvi     a,3     ; thought pattern 3
        sta     thought ;
        ret
;
load2:  lda     cdbit2
        ora     a
        jnz     xload
        lhld    comco2
        mvi     a,1     ; thought pattern 1
        sta     thought
        ret
;
load1:  lda     cdbit1
        ora     a
        jnz     xload
        lhld    comco1
        mvi     a,2     ; thought pattern
        sta     thought
        ret
;
xload:  lda     level
        inr     a
        mov     b,a
        mvi     a,3
        cmp     b
        jnc     load4
        mvi     b,1
;
load4:  mov     a,b
        sta     level
        jmp     cload
;
xload1: mvi     a,1
        sta     level
        mvi     a,0ffh
        ret
;
cstor:  lda     level
        cpi     1
        jz      stor1
        cpi     2
        jz      stor2
;
stor3:  shld    comco3
        ret
;
stor2:  shld    comco2
        ret
;
stor1:  shld    comco1
        ret
;
computr:call    cload
```

```
        cpi     0ffh
        rz
        jmp     cload1
;
computr1:
        mov     a,e
        cmp     l
        jz      compx
        jc      comup
        jmp     comdwn
;
computr2:               ; thought pattern 2
        mov     a,d
        cmp     h
        jz      computr1
        jc      comlft
        jmp     comrgt
;
computr3:               ; thought pattern 3
        mov     a,e
        cmp     l
        jz      computr2
        jc      computr4
        mov     a,e
        sui     4       ; four space miss
        cmp     l       ; test against player
        jc      comup
        jmp     comdwn
;
computr4:               ; thought pattern 3b
        mov     a,e     ; py
        adi     4
        cmp     l       ; (py+4)-cy
        jc      comup
        jmp     comdwn
;
compx:  mov     a,d
        cmp     h
        jc      comlft
;
comrgt: call    cload
        inr     h
        call    rsmem
        cpi     ' '
        jnz     comlft
        call    cstor
        mvi     a,'v'
        sta     combit
        jmp     xcomp
;
comlft: call    cload
        dcr     h
        call    rsmem
        cpi     ' '
        jnz     comup
        call    cstor
        mvi     a,'t'
        sta     combit
        jmp     xcomp
;
```

```
comup:  call    cload
        dcr     l
        call    rsmem
        cpi     ' '
        jnz     comdwn
        call    cstor
        mvi     a,'u'
        sta     combit
        jmp     xcomp
;
comdwn: call    cload
        inr     l
        call    rsmem
        cpi     ' '
        jnz     tryag
        call    cstor
        mvi     a,'s'
        sta     combit
        jmp     xcomp
;
tryag:  lda     combit4
        cpi     2
        jz      comdead
        mvi     a,2
        sta     combit4
        jmp     comrgt
;
xcomp:  call    adj      ; cursor adjustment
        call    reverse  ; reverse computer cyc's
        lda     combit
        call    grout
        call    upscrn
        call    xreverse
        lhld    oldcol
        call    adj      ; cursor adjust
        call    ssmem
        mvi     a,'i'
        call    grout
        mvi     a,1
        sta     combit4
        call    chkspd
        ora     a
        jnz     cspeed2
;
reent:  lda     lev
        mov     b,a
        lda     level
        cmp     b
        jz      xcomp1
        inr     a
        sta     level
        jmp     computr
;
cspeed2:lda     cspeed1
        cpi     4
        jnz     cspeed3
        mvi     a,2
        sta     cspeed1
        jmp     reent
;
```

```
cspeed3:mvi     a,4
        sta     cspeed1
        jmp     computr
;
xcomp1: mvi     a,1
        sta     level
        ret
;
chkspd: lda     lev1
        cpi     5
        jz      fast3
        cpi     7
        jz      fast2
        cpi     9
        jz      fast1
        lda     lev2
        ora     a
        jnz     allfast
        mvi     a,0
        ret
;
fast3:  lda     level
        cpi     3
        mvi     a,0
        rnz
        mvi     a,1
        ret
;
fast2:  lda     level
        cpi     2
        mvi     a,0
        rnz
        mvi     a,1
        ret
;
fast1:  lda     level
        cpi     1
        mvi     a,0
        rnz
        mvi     a,1
        ret
;
allfast:mvi     a,1
        ret
;
comdead:mvi     a,7
        call    output
        call    incscor
        call    comera
        call    setbit
        call    comtst
        cpi     0ffh
        jnz     computr
        call    cdmout  ; pick random sentence
        call    show
;
comded1:lda     lev1
        inr     a
        sta     lev1
        push    psw
        lda     lev1
        cpi     10
        jnz     comded2
        lda     lev2
        inr     a
        sta     lev2
        cpi     10
        jnz     comded3
        lda     lev3
        inr     a
        sta     lev3
        xra     a       ; zero second digit
        sta     lev2
;
comded3:xra     a
        sta     lev1
;
comded2:pop     psw
        cpi     3
        jc      levinc
        mvi     a,3
;
levinc: sta     lev
        cpi     1
        jz      aliv1
        cpi     2
        jz      aliv2
;
aliv3:  mvi     a,0
        sta     cdbit3
;
aliv2:  mvi     a,0
        sta     cdbit2
;
aliv1:  mvi     a,0
        sta     cdbit1
;
loop3:  call    addbonus
;
loop3b: call    wait
        cpi     cr
        jnz     loop3b
        mvi     a,1
        sta     speed1
        call    clr25
        mvi     a,0
        sta     combit
        mvi     a,1
        sta     combit4
        ret
;
setbit: lda     level
        cpi     1
        jz      setbit1
        cpi     2
        jz      setbit2
;
setbit3:mvi     a,1
        sta     cdbit3
        ret
```

```
;
setbit2:mvi     a,1
        sta     cdbit2
        ret
;
setbit1:mvi     a,1
        sta     cdbit1
        ret
;
comtst: lda     level
        cpi     1
        jz      tst1
        cpi     2
        jz      tst2
;
tst3:   lda     cdbit3
        ora     a
        rz
        jmp     xtst
;
tst2:   lda     cdbit2
        ora     a
        rz
        jmp     xtst
;
tst1:   lda     cdbit1
        ora     a
        rz
;
xtst:   lda     cdbit1
        ora     a
        rz
        lda     cdbit2
        ora     a
        rz
        lda     cdbit3
        ora     a
        rz
        mvi     a,0ffh
        ret
;
upscrn: lda     level
        cpi     1
        jz      upscr1
        cpi     2
        jz      upscr2
;
upscr3: lhld    comco3
        xchg
        lhld    screen3
        mov     m,e
        inx     h
        mov     m,d
        inx     h
        shld    screen3
        mvi     m,0
        ret
;
upscr2: lhld    comco2
        xchg
        lhld    screen2
        mov     m,e
        inx     h
        mov     m,d
        inx     h
        shld    screen2
        mvi     m,0
        ret
;
upscrl: lhld    comcol
        xchg
        lhld    screen1
        mov     m,e
        inx     h
        mov     m,d
        inx     h
        shld    screen1
        mvi     m,0
        ret
;
comera: lda     level
        cpi     1
        jz      erac1
        cpi     2
        jz      erac2
;
erac3:  lxi     h,memmap3
;
erac3a: mov     a,m
        ora     a
        rz
        mov     e,a
        inx     h
        mov     a,m
        ora     a
        rz
        mov     d,a
        inx     h
        push    h
        mov     h,d
        mov     l,e
        call    adj
        mvi     a,' '
        call    update
        call    output
        pop     h
        jmp     erac3a
;
erac2:  lxi     h,memmap2
        jmp     erac3a
;
erac1:  lxi     h,memmap1
        jmp     erac3a
;
cdmout: lxi     h,retmes; return message
        call    show
        call    spmsg   ; Stack Pointer Message added  04/06/83
        db      '[S1]@',25,1,'[c]',0
        lhld    clock   ; get clock counter
        mov     a,m     ; get counter
        cpi     0ah     ; less than 10 ?
        lxi     h,cdmes1
        jc      show
        cpi     14h
        lxi     h,cdmes2
        jc      show
        cpi     1eh
        lxi     h,cdmes3
        jc      show
        cpi     28h
        lxi     h,cdmes4
        jc      show
        cpi     32h
        lxi     h,cdmes5
        jc      show
        cpi     3ch
        lxi     h,cdmes6
        jc      show
        cpi     46h
        lxi     h,cdmes7
        jc      show
        cpi     50h
        lxi     h,cdmes8
        jc      show
        cpi     5ah
        lxi     h,cdmes9
        jc      show
        cpi     64h
        lxi     h,cdmes10
        jc      show
        cpi     6eh
        lxi     h,cdmes11
        jc      show
        cpi     78h
        lxi     h,cdmes12
        jc      show
        cpi     0beh     ; 190
        lxi     h,cdmes13
        jc      show
        cpi     0d2h     ; 210
        lxi     h,cdmes14
        jc      show
        lxi     h,cdmes15
        jmp     show
;
cdmes1: db      ' ** You got lucky!! If you think you''re so good'
        db      ' increase the level.',0
;
cdmes2: db      ' ** Alright you, you''re getting me angry!!  I''ll'
        db      ' have to send my best warriors!!!',0
;
cdmes3: db      ' ** %@#&$ User''s!!!!  You''ll regret this.....',0
;
cdmes4: db      ' ** Well, I didn''t say I had the BEST warriors.',0
;
cdmes5: db      ' ** That was one of my worst warriors, try this one...',0
;
cdmes6: db      ' ** Good move -- see, I''m not a poor sport...',0
;
cdmes7: db      ' ** Sooo, you say you''re good huh??  Well, the next time'
```

```
        db      ' you won''t be so fortunate.',0
;
cdmes8: db      ' ** ** ** ** NO COMMENT ** ** ** **',0
;
cdmes9: db      ' ?*?*?*  What happened *?*?*?   My warrior must''ve slipped'
        db      '...',0
;
cdmes10:db      ' ** This is you''re boss here....STOP THAT!!!',0
;
cdmes11:db      ' ** OUCH !!!!  That one was part of my I/O program.',0
;
cdmes12:db      ' ** Do you take bribes???  You''re making me look bad!!!',0
;
cdmes13:db      ' ** Hmmmmmm.....There must''ve been oil on the grid..',0
;
cdmes14:db      ' ** I''m going to QUIT if you keep that up....',0
;
cdmes15:db      ' ** LUCK !!!',0
;
pdmes:  db      '@',25,1,'[c] Try again.....',0
;
retmes: db      '@',24,1,'[c]@',24,34,'PRESS [R] RETURN [r]',0
;
cload1: shld    oldcol  ; save for wall
        xchg            ; computer x,y
        lhld    playco  ; player x,y
        xchg            ; DE=px,py  HL=cx,cy
        push    h
        push    d
        lda     lev3    ; high level bit
        ora     a
        jz      cload2  ; high bit = 0
        adi     09h     ; add 9
;
cload2: mov     b,a     ; store in B
        lda     lev2    ; low level bit
        add     b       ; add reg. B
        cpi     28      ; highest level
        jc      cload3  ; subtract 27
        sui     1bh     ;
;
cload3: mov     b,a     ; store in B
        mvi     c,6     ; multiply by 7
        xra     a       ; clear accumalator
;
cload4: add     b       ;
        dcr     c
        jnz     cload4
        mov     e,a
        lda     thought
        dcr     a
        add     a       ; double 1=2,2=4
        add     e
        mvi     d,0     ;
        mov     e,a     ; put in E
        lxi     h,patterns
        dad     d
        mov     c,m
        inx     h       ; next location
        mov     b,m
        pop     d
        pop     h
        push    b
        ret             ; return to selected move
;
patterns:
        dw      computr1,computr1,computr1      ; 1
        dw      computr2,computr1,computr1      ; 2
        dw      computr3,computr1,computr1      ; 3
        dw      computr1,computr1,computr2      ; 4
        dw      computr2,computr1,computr2      ; 5
        dw      computr3,computr1,computr2      ; 6
        dw      computr1,computr1,computr3      ; 7
        dw      computr2,computr1,computr3      ; 8
        dw      computr3,computr1,computr3      ; 9
        dw      computr1,computr2,computr1      ; 10
        dw      computr2,computr2,computr1      ; 11
        dw      computr3,computr2,computr1      ; 12
        dw      computr1,computr2,computr2      ; 13
        dw      computr2,computr2,computr2      ; 14
        dw      computr3,computr2,computr2      ; 15
        dw      computr1,computr2,computr3      ; 16
        dw      computr1,computr3,computr1      ; 17
        dw      computr1,computr3,computr2      ; 18
        dw      computr1,computr3,computr3      ; 19
        dw      computr3,computr2,computr3      ; 20
        dw      computr3,computr3,computr1      ; 21
        dw      computr3,computr3,computr2      ; 22
        dw      computr3,computr3,computr3      ; 23
        dw      computr2,computr2,computr3      ; 24
        dw      computr2,computr3,computr1      ; 25
        dw      computr2,computr3,computr2      ; 26
        dw      computr2,computr3,computr3      ; 27
;
init:   lxi     h,2814h
        shld    playco
        lxi     h,2806h
        shld    comco
        shld    memmap1
        lxi     h,2106h
        shld    comco2
        shld    memmap2
        lxi     h,2f06h
        shld    comco3
        shld    memmap3
        mvi     a,'u'
        sta     pldir
        mvi     a,'8'
        sta     plmove
        mvi     a,'s'
        sta     combit
        sta     combit2
        sta     combit3
        lxi     h,memmap1+2
        shld    screen1
        lxi     h,memmap2+2
        shld    screen2
        lxi     h,memmap3+2
        shld    screen3
        ret
```

```
misc:   lxi     h,1402h
        call    adj
        lxi     h,score1
        call    show
        lxi     h,4302h
        call    adj
        call    levout
        lxi     h,4301h
        call    adj
        call    levout1
        lxi     h,2001h
        call    adj
        lxi     h,hscore1
        call    show
        lxi     h,2801h
        call    adj
        lxi     h,name1
        call    show
        lxi     h,1f02h
        call    adj
        call    cycles
        ret
;
thought:db      0       ; thought pattern storage
;
cycles: lda     cycleft ; don't display
        dcr     a       ; 1st cycle in use
        rz
cycles1:push    psw
        mvi     a,'v'
        call    grout
        pop     psw
        dcr     a
        jnz     cycles1
        ret
;
levout: lda     lev1
        adi     30h
        call    output
        ret
;
levout1:lda     lev2
        inr     a
        cpi     0ah
        jc      levout2
        lda     lev3
        inr     a
        adi     30h
        call    output
        xra     a
levout2:adi     30h
        call    output
        ret
;
dislevel:
        lda     lev1    ; see if time to display
        mov     b,a
        lda     lev2
        add     b
        cpi     1       ; test for level 0, round 1
        jz      dislevel1
        lda     lev1
        ora     a
        rnz
dislevel1:
        call    graphix ; enter graphics mode
        lxi     h,levmes; level message
        call    show
dislevel2:
        lxi     h,370ah
        call    adj     ; display level number
        lda     lev3
        cpi     0       ; 1st digit 0?
        jz      dislevel3
        call    disslect1
        lxi     h,3e0ah
        call    adj     ; next digit position
dislevel3:
        lda     lev2
        inr     a
        call    disslect1
        call    xgraphix; exit graphics mode
        lxi     h,1500h ; delay
disslect:
        dcx     h
        push    h
        call    input   ; test for input
        pop     h
        ora     a       ; test
        rnz
        mov     a,h
        ora     l
        jnz     disslect
        ret
;
disslect1:
        cpi     0       ; test for zero
        jz      diszero
        cpi     1
        jz      disone
        cpi     2
        jz      distwo
        cpi     3
        jz      disthree
        cpi     4
        jz      disfour
        cpi     5
        jz      disfive
        cpi     6
        jz      dissix
        cpi     7
        jz      disseven
        cpi     8
        jz      diseight
disnine:lxi     h,nine
        call    show
        ret
diseight:
        lxi     h,eight
        call    show
        ret
disseven:
        lxi     h,seven
        call    show
        ret
dissix: lxi     h,six
        call    show
        ret
disfive:lxi     h,five
        call    show
        ret
disfour:lxi     h,four
        call    show
        ret
disthree:
        lxi     h,three
        call    show
        ret
distwo: lxi     h,two
        call    show
        ret
disone: lxi     h,one
        call    show
        ret
diszero:lxi     h,zero
        call    show
        ret
;
incscor:lxi     h,score1
        inx     h
        inx     h
        inx     h       ; get 100's score
        mov     b,m
        lda     lev
        add     b
        cpi     ':'
        jnc     inc2
        mov     m,a
        jmp     scorout
inc2:   sui     10
        mov     m,a
        dcx     h
        mov     a,m
        inr     a
        cpi     ':'
        jnz     xscor
        jmp     inc2
xscor:  mov     m,a
scorout:lxi     h,1402h
        call    adj
        lxi     h,score1
        call    show
        lhld    freepnt
        lxi     d,score1
        inx     d       ; 10,000's digit
        ldax    d
        cmp     m
        rnz
        inx     h
        inx     d
        ldax    d
        cmp     m
        rc
        inx     h
        shld    freepnt
        mov     a,m
        cpi     0
        jnz     bonus
        lxi     h,freebie
        shld    freepnt
bonus:  lda     cycleft
        inr     a
        sta     cycleft
sound:  mvi     a,07h
        call    output
        mvi     a,07h
        call    output
        mvi     a,07h
        call    output
        ret
pause:  call    wait
        jmp     game1
newhigh:lxi     h,3100h
        shld    hscore6
        lxi     h,hscore1
nwhigh: lxi     d,pscore1
        ldax    d
        cmp     m       ; 100,000 position
        jc      nxthigh
        jnz     nhigh1
nhigh2: inx     h
        inx     d
        ldax    d
        cmp     m       ; all other digits
        jc      nxthigh
        jnz     nhigh1  ;
        jmp     nhigh2
nhigh1: lxi     h,nhmes ; new high message
        call    show
        lxi     h,1e11h
        call    adj     ; display position
        lda     hscore6+1
        call    output
        call    pushdown; push lower scores down.
        call    curhigh
        lxi     d,pscore1
nhigh3: ldax    d
        mov     m,a
        inx     h
        inx     d
        mov     a,m
        ora     a
        jnz     nhigh3  ; repeat until all digits copied
        lxi     h,nhmes1
        call    show
        call    nhnames
        call    zeroname
        call    curson
        call    cursb
        mvi     m,32
        inx     h
        call    inbuf
        inx     h
        mvi     m,0
        call    cursoff
        ret
zeroname:
        push    h
        mvi     b,0eh   ; 14 char max
zeronam1:
        mvi     m,32
        inx     h
        dcr     b
        jnz     zeronam1
        pop     h
        ret
nxthigh:call    nxthigh1
        jmp     nwhigh
nxthigh1:
        lhld    hscore5
        inr     h
        shld    hscore6
curhigh:lhld    hscore6
        mov     a,h
        sui     31h
        cpi     0
        jz      hscr1
        cpi     1
        jz      hscr2
        cpi     2
        jz      hscr3
        cpi     3
        jz      hscr4
        cpi     4
        jz      hscr5
        cpi     5
        jz      hscr6   ; exit
        ret             ; return to gameov
hscr1:  lxi     h,hscore1
        ret
hscr2:  lxi     h,hscore2
        ret
hscr3:  lxi     h,hscore3
        ret
hscr4:  lxi     h,hscore4
        ret
hscr5:  lxi     h,hscore5
        ret
hscr6:  pop     h       ; get return address
        ret
nhnames:lhld    hscore6
        mov     a,h
        sui     31h
        cpi     1
        jz      nhname2
        cpi     2
        jz      nhname3
        cpi     3
        jz      nhname4
        cpi     4
        jz      nhname5
nhname1:lxi     h,name1+2
        ret
nhname2:lxi     h,name2+2
        ret
nhname3:lxi     h,name3+2
        ret
nhname4:lxi     h,name4+2
        ret
nhname5:lxi     h,name5+2
        ret
highscore:
        lxi     h,chart
        call    show
        lxi     h,chart2
        call    show
        lxi     h,1405h
        call    adj
        lxi     h,name1
        call    show
        lxi     h,3205h
        call    adj
        lxi     h,hscore1
        call    show
        lxi     h,1407h
        call    adj
        lxi     h,name2
        call    show
        lxi     h,3207h
        call    adj
        lxi     h,hscore2
        call    show
        lxi     h,1409h
        call    adj
        lxi     h,name3
        call    show
        lxi     h,3209h
        call    adj
        lxi     h,hscore3
        call    show
        lxi     h,140bh
        call    adj
        lxi     h,name4
        call    show
        lxi     h,320bh
        call    adj
        lxi     h,hscore4
        call    show
        lxi     h,140dh
        call    adj
        lxi     h,name5
        call    show
        lxi     h,320dh
        call    adj
        lxi     h,hscore5
        call    show
highscore1:
        call    reverse
        lxi     h,chart1
        call    show
        call    xreverse
        call    input
        ora     a
        jnz     begin
        lxi     h,chart1
        call    show
        call    input
        ora     a
        jnz     begin
        jmp     highscore1
pushdown:
        lhld    hscore6 ; get counter
        mov     a,h     ;
        sui     30h     ; offset
        cpi     1       ; top score?
        jz      pushall ; push all scores
        cpi     2       ; second highest
        jz      push2d  ;
        cpi     3       ; third highest
        jz      push3d  ;
        cpi     4       ; fourth place
        jz      push4d  ;
        ret             ; replace 5th
push1:  call    nhname1 ; get 1st name
        lxi     d,name2 ; destination
        call    copyname; copy it
        lxi     h,hscore1
        lxi     d,hscore2
        call    copyscor;
        ret
push2:  call    nhname2 ; name to copy
        lxi     d,name3 ; where to copy
        call    copyname;
        lxi     h,hscore2
        lxi     d,hscore3
        call    copyscor;
        ret
push3:  call    nhname3 ;
        lxi     d,name4 ;
        call    copyname;
        lxi     h,hscore3
        lxi     d,hscore4
        call    copyscor;
        ret
push4:  call    nhname4 ;
        lxi     d,name5 ;
        call    copyname;
        lxi     h,hscore4
        lxi     d,hscore5
        call    copyscor;
        ret
copyname:
        mov     a,m     ; get first char
        stax    d       ; store it
        inx     d       ;
        inx     h       ;
        ora     a       ; test after copied
        rz              ; return if zero
        jmp     copyname; else loop.
copyscor:
```

```
copyscl:mov     a,m      ;
        ora     a        ; test for delimiter
        rz               ; return if zero
        stax    d        ; store score
        inx     d        ;
        inx     h        ;
        jmp     copyscl  ;
pushall:call    push4    ;
        call    push3    ;
        call    push2    ;
        call    push1    ;
        ret
push2d: call    push4    ;
        call    push3    ;
        call    push2    ;
        ret
push3d: call    push4    ;
        call    push3    ;
        ret
push4d: call    push4    ;
        ret
copyright:
        lxi     h,1717h  ; 24th line, 24th column
        call    adj      ; put cursor there
        lda     cbit     ;
        cpi     1        ;
        jnz     cpyrgt1  ;
cpyrgt2:lxi     h,copyrgt
        call    show     ; print message
        mvi     a,0
        sta     cbit
        ret
cpyrgt1:call    reverse  ;
        lxi     h,copyrgt
        call    show     ;
        mvi     a,1
        sta     cbit
        jmp     xreverse
;
; game grid
;
grid:   db      '[CG]@',3,1,'l{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{'
        db      '{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{m'
        db      '@',4,1,'}@',4,80,'|@',5,1,'}@',5,80,'|'
        db      '@',6,1,'}@',6,80,'|@',7,1,'}@',7,80,'|'
        db      '@',8,1,'}@',8,80,'|@',9,1,'}@',9,80,'|'
        db      '@',10,1,'}@',10,80,'|@',11,1,'}@',11,80,'|'
        db      '@',12,1,'}@',12,80,'|@',13,1,'}@',13,80,'|'
        db      '@',14,1,'}@',14,80,'|@',15,1,'}@',15,80,'|'
        db      '@',16,1,'}@',16,80,'|@',17,1,'}@',17,80,'|'
        db      '@',18,1,'}@',18,80,'|@',19,1,'}@',19,80,'|'
        db      '@',20,1,'}@',20,80,'|@',21,1,'}@',21,80,'|'
        db      '@',22,1,'}@',22,80,'|'
        db      '@',23,1,'ozzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz'
        db      'zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzn[g]'
        db      '@',2,10,' SCORE  :@',1,60,'Level:@',2,60,'Round:'
        db      '@',1,16,'HIGH SCORE :'
        db      '@',24,24,'Press : (Q) - QUIT / (P) - PAUSE',0
;
box1:   lxi     h,490fh ; opposite corner
```

```
        shld    boxcol
        lxi     h,708h  ; line 8 column 7
box2:   shld    boxco
        call    box3
        lhld    boxcol
        push    h
        call    box3
        pop     h
        mov     a,h
        cpi     7
        rz
        dcr     h
        shld    boxcol
        lhld    boxco
        inr     h
        jmp     box2    ; print next block
box7:   lhld    boxco
box4:   shld    boxco
        call    box3
        lhld    boxcol
        push    h
        call    box3
        pop     h
        mov     a,l     ; get y
        cpi     8
        rz
        dcr     l
        shld    boxcol
        lhld    boxco
        inr     l
        jmp     box4
box3:   call    adj
        call    reverse
        lda     boxchar
        call    grout
        call    xreverse
        ret
intro:  db      '[CG]@',2,32,'fa asaa faaaaaaaaa'
        db      '@',3,32,'`   `   eac@',4,32,'eaaadaaaaad'
        db      '@',3,33,'[g]omp@',3,37,'ec@',3,43,'oftware'
        db      '@',6,37,'presents[G]@',9,13,'{{{{@',9,31,'{'
        db      '@',9,44,'{{ {{@',9,61,'lm@',10,12
        db      'x@',10,25,'{{{{ } |   {{{    x| y| y   {{{  {{'
        db      '     {{{@',11,12,'|       x|   x      w   x'
        db      '{{{|     |  |  | x   | | y}| x   |'
        db      '@',12,12,'y{{{{{x y{{xy{{{{{x y{{w{{{{{   '
        db      '|     yxy{{xyx| }xyxy{{xy@',13,13,'{{{{{{{{{{x[g]'
        db      '@',16,39,'by@',18,36,'Les Bird@',2,62,'SKILL LEVEL:'
        db      '@',16,9,'Press [R](1)[r] to play CYCLE@',18,15
        db      '[R](2)[r] to see high''s@',16,49
        db      'Press [R](3)[r] to select options@',18,49
        db      '       [R](5)[r] to quit',0
chart:  db      '[C]@',2,29,'[R] C Y C L E  M A N I A [r]'
        db      '@',3,30,'ALL TIME HIGH SCORES'
chart1: db      '@',4,18,'[G]faaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
        db      'aaaaaaaaaaaac@',5,18,''@',5,62,''
        db      '@',6,18,''@',6,62,''@',7,18,''@',7,62
        db      ''@',8,18,''@',8,62,''@',9,18,''@',9,62
        db      ''@',10,18,''@',10,62,''@',11,18,''
        db      '@',11,62,''@',12,18,''@',12,62,''
        db      '@',13,18,'`@',13,62,'`@',14,18
        db      'eaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaad[g]',0
chart2: db      '@',5,16,'1.@',7,16,'2.@',9,16,'3.@',11,16,'4.'
        db      '@',13,16,'5.',0
mess10: db      '[CR]@',2,25,' C Y C L E  M A N I A [r]'
        db      '[r]@',5,20,'Select speed level :'
        db      ' [R](1)[r] for expert@',6,41,'[R](2)[r] for intermediate'
        db      '@',7,41,'[R](3)[r] for level 1 intermediate@',8,41
        db      '[R](4)[r] for novice',0
nhmes:  db      '[CG]@',8,18
        db      esc,'jfaac',esc,'k',lf,'`',bs,lf,'eaad@',9,23   ; C
        db      esc,'jfc',esc,'k',lf,'ed@',9,26                 ; o
        db      esc,'jsc',esc,'k',lf,'ee@',9,29                 ; n
        db      esc,'jfc',esc,'k',lf,'et@',11,18,'aaaaaaaaaaaad'; g
        db      '@',9,32
        db      esc,'jfa',esc,'k',lf,'e@',9,35                  ; r
        db      esc,'jfc',esc,'k',lf,'eu@',8,38                 ; a
        db      'f@',9,37,'aba@',10,38,'e@',9,41                ; t
        db      esc,'jff',esc,'k',lf,'ed@',8,44                 ; u
        db      esc,'jc',esc,'k',lf,'`@',10,44,'e@',9,46        ; l
        db      esc,'jfc',esc,'k',lf,'eu@',8,50                 ; a
        db      'f@',9,49,'aba@',10,50,'e@',9,53                ; t
        db      esc,'js',esc,'k',lf,'u@',9,55                   ; i
        db      esc,'jfc',esc,'k',lf,'ed@',9,58                 ; o
        db      esc,'jsc',esc,'k',lf,'ee@',9,61                 ; n
        db      esc,'jf',esc,'k',lf,'d[g]'                      ; s
        db      '@',14,21,'You made the ALL TIME HIGH SCORE CHART'
        db      '@',15,22,'Enter your name, 14 character limit.'
nhmes1: db      '@',17,31,'.______________@',17,32,0
;
expmes: db      '@',3,64,'[G]l[R]pppppp[r]m@',4,64,'[R]q[rg]EXPERT'
        db      '[G]q@',5,64,'opppppppn[g]',0
promes: db      '@',3,65,'[G]{{{{{[R]@',4,65,'q[r]PROq@',5,66,'zzzzz[g]',0
intmes: db      '@',3,62,'[G] {{{{{{{{{{{{ @',4,62,'}INTERMEDIATE|'
        db      '@',5,62,' zzzzzzzzzzzz [g]',0
amames: db      '@',3,62,'@',4,64,'AMATUER@',5,64,0
copyrgt:db      '(C)opyright 1983 CompTec Service',0
;
cycle19:call    cls
        call    clr25
cycle0: lxi     h,4f19h ; column 79, line 25
        call    adj
        mvi     a,25
        sta     misbit
        lxi     h,cyclemes
cycle1: mov     a,m
        ora     a
        jz      cycle0
        push    h
        call    output
        lxi     h,119h  ; x=1, y=25
        call    adj
        call    deletec
        lxi     h,4f19h
        call    adj
        lxi     h,2500h
cycle2: dcx     h
        mov     a,h
        ora     l
        jnz     cycle2
        call    input
        ora     a
        pop     h
        jnz     cycle3
        inx     h
        jmp     cycle1
cycle3: call    clr25
        jmp     begin
;
; open or create high score file
;
highfile:
        call    openfile; open the file
        call    readfile; read data
        ora     a
        rnz
; read high scores from disk
        lxi     b,0580h ; five high scores
        lxi     h,dma
high1:  lxi     d,name1 ; copy name *** storage location
highfile1:
        mov     a,m
        dcr     c
        cz      highfile6
        cpi     '='     ; name/score separator
        jz      highfile2
        stax    d
        inx     h
        inx     d
        jmp     highfile1
highfile2:      ; ********** storage location **********
        lxi     d,hscore1
        inx     h
highfile3:
        mov     a,m     ; get score
        dcr     c
        cz      highfile6
        cpi     '/'     ; load next name/score
        jz      highfile4
        cpi     '*'     ; end of file
        rz
        stax    d       ; store in score
        inx     h
        inx     d
        jmp     highfile3
highfile4:
        inx     h       ; start of next name
        dcr     b       ; count=count-1
        rz
        mov     a,b
        cpi     1
        jz      high5
        cpi     2
        jz      high4
        cpi     3
        jz      high3
high2:  push    h
        lxi     h,name2
        shld    high1+1
        lxi     h,hscore2
        shld    highfile2+1
        pop     h
        jmp     high1
high3:  push    h
        lxi     h,name3
        shld    high1+1
        lxi     h,hscore3
        shld    highfile2+1
        pop     h
        jmp     high1
high4:  push    h
        lxi     h,name4
        shld    high1+1
        lxi     h,hscore4
        shld    highfile2+1
        pop     h
        jmp     high1
high5:  push    h
        lxi     h,name5
        shld    high1+1
        lxi     h,hscore5
        shld    highfile2+1
        pop     h
        jmp     high1
highfile6:
        call    savall
        call    readfile
        call    retall
        mvi     c,80h
        lxi     h,dma
        ret
; write high scores to disk
wrhigh: call    delfile
        call    makefile
        lxi     d,dma
        lxi     b,0580h ; five high scores
whigh6: lxi     h,name1 ; **** storage location ****
wrhigh1:mov     a,m
        stax    d
        dcr     c
        cz      wrhigh6
        ora     a
        jz      wrhigh2
        inx     h
        inx     d
        jmp     wrhigh1
wrhigh2:inx     d
        mvi     a,'='   ; name/score separator
        stax    d       ; put in DMA
        dcr     c
        cz      wrhigh6
        inx     d
whigh7: lxi     h,hscore1
wrhigh3:mov     a,m
        stax    d
        dcr     c
        cz      wrhigh6
        ora     a
        jz      wrhigh4
        inx     h
        inx     d
        jmp     wrhigh3
wrhigh4:mvi     a,'/'   ; *** storage location ***
        inx     d
        stax    d
        dcr     b
        jz      wrhigh7
        dcr     c
        cz      wrhigh6
        inx     d
        mov     a,b
        cpi     1
        jz      whigh5
        cpi     2
        jz      whigh4
        cpi     3
        jz      whigh3
whigh2: lxi     h,name2
        shld    whigh6+1
        lxi     h,hscore2
        shld    whigh7+1
        jmp     whigh6
whigh3: lxi     h,name3
        shld    whigh6+1
        lxi     h,hscore3
        shld    whigh7+1
        jmp     whigh6
whigh4: lxi     h,name4
        shld    whigh6+1
        lxi     h,hscore4
        shld    whigh7+1
        jmp     whigh6
whigh5: lxi     h,name5
        shld    whigh6+1
        lxi     h,hscore5
        shld    whigh7+1
        mvi     a,'*'   ; end of file
        sta     wrhigh4+1
        jmp     whigh6
wrhigh6:call    savall  ; all registers on stack
        call    writefile
        call    retall
        mvi     c,80h   ; another 128 bytes
        lxi     d,dma   ; reset DMA
        ret
wrhigh7:dcr     c
        jz      wrhigh6 ; write file
        mvi     a,20h   ; space
        inx     d
        stax    d
        jmp     wrhigh7
cyclemes:
        db      'Greetings program,  I am the Master of this program'
        db      ' and I have a challenge for you....I have cyclist'
        db      ' who would like to go against you.   And if you feel'
        db      ' that you can beat them, just PRESS ANY KEY on your'
        db      ' keyboard and I will transport you to the game grid.'
        db      '  I was written by:  Les Bird  1983 CompTec Service'
        db      '                            END OF LINE.           '
        db      '----)       C Y C L E   M A N I A       (----    ',0
```

```
cyclemes1:
        db      'CYCLE MANIA',0
levmes: db      '[C]@',10,19
        db      '[R]r[r]r    [R]r[r]ppp[R]_[r] [R]r[r]    [R]_[r]'
        db      ' [R]r[r]ppp[R]_[r] [R]r[r]r'
        db      '@',11,19
        db      '[R] [r]     [R] [r]     _[R]_[r]  [R]r[r]r '
        db      '[R] [r]     [R] [r]'
        db      '@',12,19
        db      '[R] [r]     [R] [r]ppp   _[R]_r[r]r  '
        db      '[R] [r]ppp  [R] [r]'
        db      '@',13,19
        db      '_[R]ppp[r]r _[R]ppp[r]r   _r   _[R]ppp[r]r _[R]ppp[r]r',0
levmes1:db      '@',8,28,'[g]Enter level: 1=easy, 9=hard[G]',0
one:    db      1bh,'j [R]r [rc]',1bh,'k',1f
        db      1bh,'j  [R] [rc]',1bh,'k',1f
        db      1bh,'j  [R] [rc]',1bh,'k',1f
        db      ' [R]p p[r]    ',0
two:    db      1bh,'j[R]r[r]pppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j [R]pppp[r]r[c]',1bh,'k',1f
        db      1bh,'j[R]r[rc]',1bh,'k',1f
        db      '[R] pppp[rc]',0
three:  db      1bh,'j[R]r[r]pppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j  {{{[R] [rc]',1bh,'k',1f
        db      1bh,'j  zzz[R] [rc]',1bh,'k',1f
        db      '_[R]pppp[r]r[c]',0
four:   db      1bh,'j[R]r[r]   [R]_[rc]',1bh,'k',1f
        db      1bh,'j[R] [r]   [R] [rc]',1bh,'k',1f
        db      1bh,'jpppp[R] [r]p[c]',1bh,'k',1f
        db      '    [R] [rc]',0
five:   db      1bh,'j[R] [r]ppppp[c]',1bh,'k',1f
        db      1bh,'j[R] ppppp[rc]',1bh,'k',1f
        db      1bh,'j     [R] [rc]',1bh,'k',1f
        db      '[R]ppppp[r]r[c]',0
six:    db      1bh,'j[R]r[r]pppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j[R] [r]     [c]',1bh,'k',1f
        db      1bh,'j[R] [r]pppp[R]_[rc]',1bh,'k',1f
        db      '_[R]pppp[r]r[c]',0
seven:  db      1bh,'j[R]r[r]ppp[R] [r]r[c]',1bh,'k',1f
        db      1bh,'j   [R]r[r]r[c]',1bh,'k',1f
        db      1bh,'j  [R]r[r]r[c]',1bh,'k',1f
        db      ' [R]r[r]r[c]',0
eight:  db      1bh,'j[R]r[r]pppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j_{{{{r[c]',1bh,'k',1f
        db      1bh,'j[R]r[r]zzzz[R]_[rc]',1bh,'k',1f
        db      '_[R]pppp[r]r[c]',0
nine:   db      1bh,'j[R]r[r]pppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j_[R]pppp [rc]',1bh,'k',1f
        db      1bh,'j     [R] [rc]',1bh,'k',1f
        db      '_[R]pppp[r]r[c]',0
zero:   db      1bh,'j[R]r [r]ppp[R]_[rc]',1bh,'k',1f
        db      1bh,'j[R] [r]_[R]_[r]  [R] [rc]',1bh,'k',1f
        db      1bh,'j[R] [r]  _[R]_ [rc]',1bh,'k',1f
        db      '_[R]ppp [r]r[c]',0
;
; error messages
;
nserror:db      '@',12,5,'[r]I think you should allocate more disk space.',0
nferror:db      '@',12,5,'[r]It seems I am trying to open a file that'
        db      ' does not exist.',0
```

```
rerror: db      '@',12,5,'[r]In order for me to read from this disk, you'
        db      ' must BOOT UP on it.',0
wrerror:db      '@',12,5,'[r]I think you need to BOOT UP on this disk so'
        db      ' that I can write on it.',0
;
; disk equates
;
drive   db      0        ; select current drive
extnum  db      0
reccnt  db      0
currec  db      0
recnum  db      0
filename:
        db      'CYCLE19 ',0
filetype:
        db      'DAT',0
;
retvec: equ     0674h
gbit:   equ     0677h
rbit:   equ     0678h
cbit:   db      0
misbit: db      0
crtbit: equ     0679h
graphx: equ     067ah
kpad:   equ     067bh
;                B    D    E    C   ; Free cycle soundfx
sndmem: db      002h,001h,00fh,06fh ; 1st sound
;
oldstack:equ     068ah
stack:  equ     oldstack+64h
cycleft:db      4
char1:  db      0
misco:  ds      2
curco:  equ     067ch
curco1: db      1,1
boxco:  equ     0682h
boxco1: equ     0684h
boxchar:db      'i'
linco:  equ     067eh
linco1: equ     0680h
linchar:db      'i'
placo:
playco:
playco1:db      1,1
cntrls: db      0
pldir:
plabit:
plabit1:ds      1
plmove:
plamove:
plmove1:db      '8'
speed:
speed1: db      1
speed2: db      1
speed3: db      1
comco:
comco1: db      1,1
comco2: db      1,1
comco3: db      1,1
comdir:
```

```
combit:
combit1:db      1
combit2:db      1
combit3:db      1
combit4:db      1
cdbit1: db      0
cdbit2: db      1
cdbit3: db      1
cspeed1:db      0
oldco:
oldco1: db      1,1
wall:
wall1:  ds      1
pscore1:
score:
score1: db      '000000',0
names:
name1:  db      ' ',1,'[R]C[r] o m p [R]T[r] e c ',0
name2:  db      ' ',1,'[R]C[r] o m p [R]T[r] e c ',0
name3:  db      ' ',1,'[R]C[r] o m p [R]T[r] e c ',0
name4:  db      ' ',1,'[R]C[r] o m p [R]T[r] e c ',0
name5:  db      ' ',1,'[R]C[r] o m p [R]T[r] e c ',0
hscore:
hscore1:db      '015000',0
hscore2:db      '013000',0
hscore3:db      '009000',0
hscore4:db      '005000',0
hscore5:db      '001200',0
hscore6:ds      2
freepnt:ds      2
; FREEBIE is list of free cycle points
; ex. 15 = 15,000/30 = 30,000 etc.
freebie:db      '153045607590',0
time:   equ     0586h
level:  db      1
lev:    db      1
lev1:   db      1
lev2:   db      0
lev3:   db      0
skill:  db      3
scrpnt: equ     0588h
screen1:ds      2
screen2:ds      2
screen3:ds      2
screen4:ds      2
memmap1:ds      1000
memmap2:ds      1000
memmap3:ds      1000
memmap: equ     05f4h
;
        end     start
```