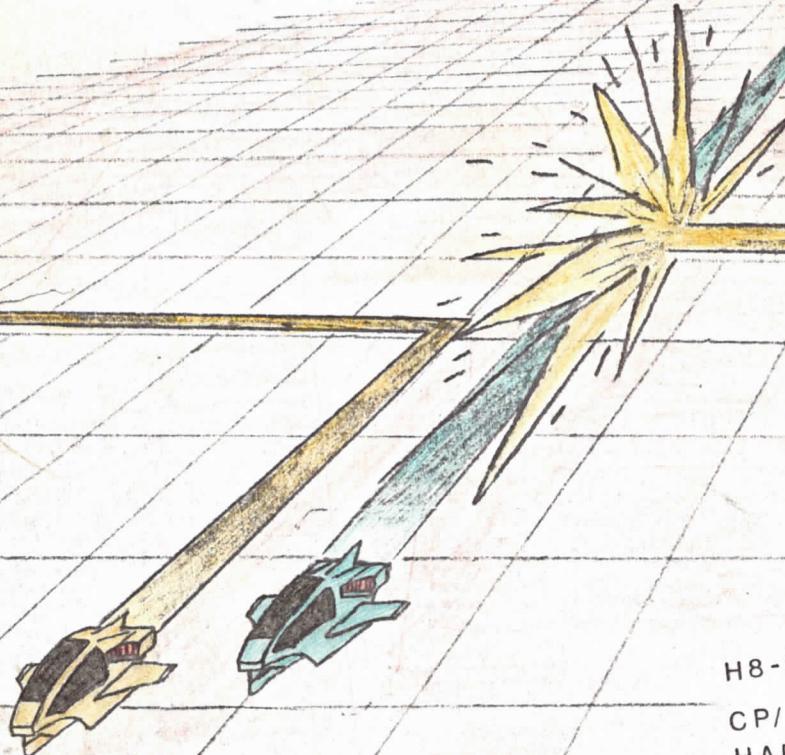


**COMPTEC  
SOFTWARE**

**PRESENTS**

# **CYCLEMANIA**

**BY LES BIRD**



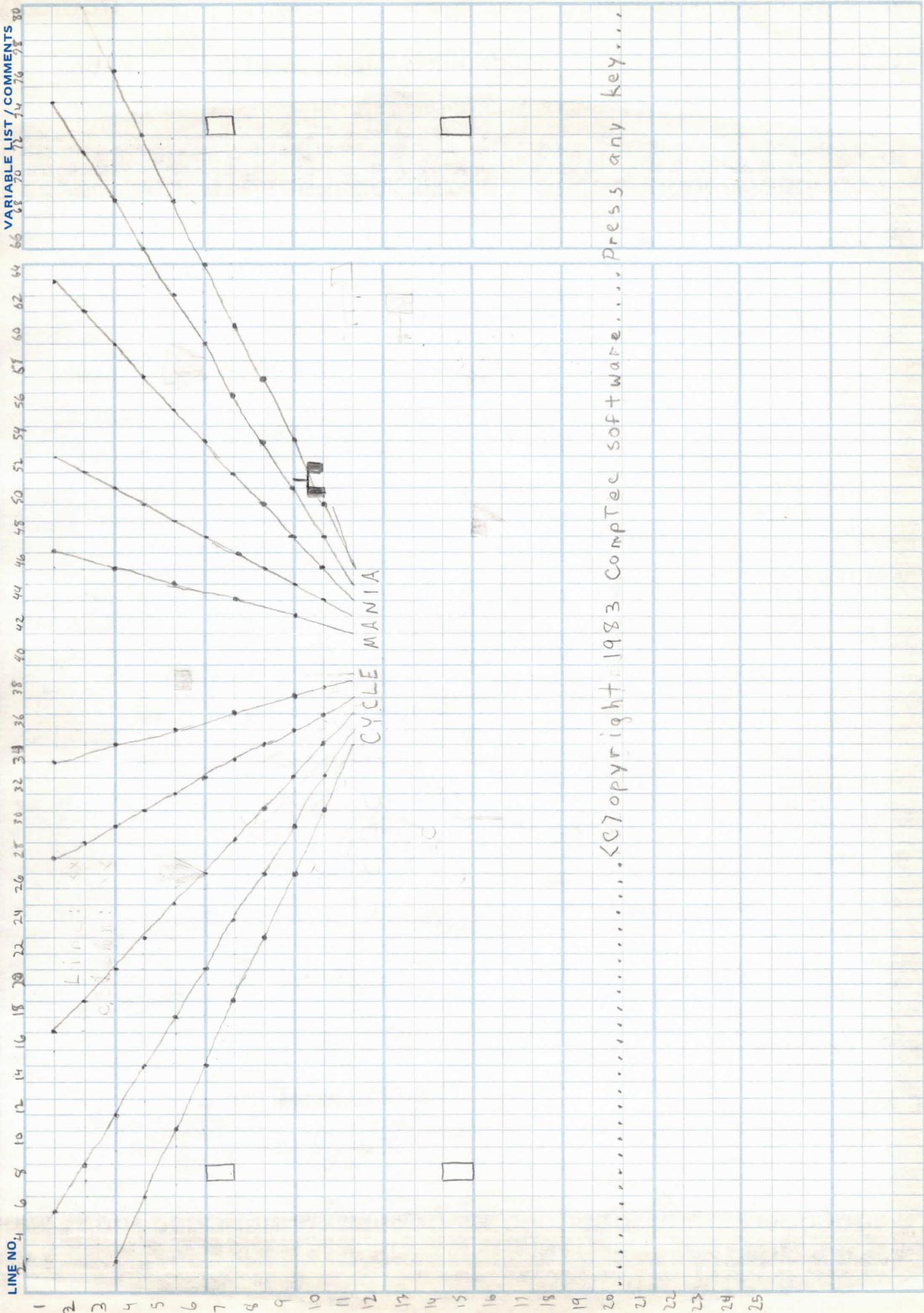
**H8-H19/H89**

**CP/M  
HARD SECT**

## COMMENTS

## PROGRAMMER

## PAGE OF



Copyright 1983 Comptec Software... Press any key...  
20  
21  
22  
23  
24  
25

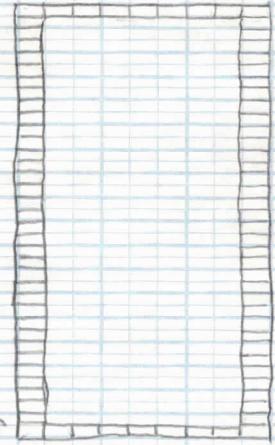
# TRS-80 Video Display Worksheet

**Radio Shack®**

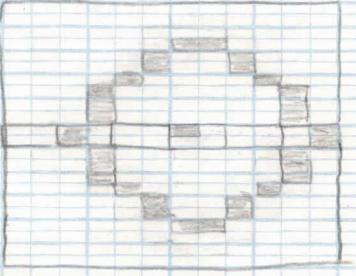
TITLE \_\_\_\_\_ PROGRAMMER \_\_\_\_\_ PAGE \_\_\_\_ OF \_\_\_\_

X  
Y  
Z

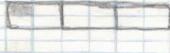
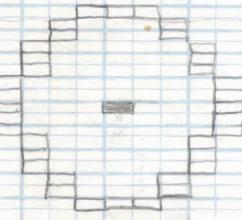
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
PRINT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
AT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Z	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47



(97,37)



(62,30)



# TRON 80

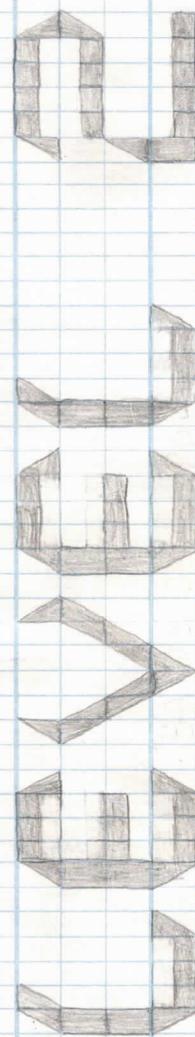
BASIC Coding Form

RadioS tek

BONUS 1000 POINTS

TITLE	LINE NO.	PROGRAMMER	COMMENTS	PAGE OF
1	19			68 VARIABLE LIST / COMMENTS
2	1			68
3	2			69
4	3			70
5	4			71
6	5			72
7	6			73
8	7			74
9	8			75
10	9			76
11	10			77
12	11			78
13	12			79
14	13			80
15	14			81
16	15			82
17	16			83
18	17			84
19	18			85
20	19			86
21	20			87
22	21			88
23	22			89
24	23			90
25	24			91

Enter level: 1=easy, 9=hard

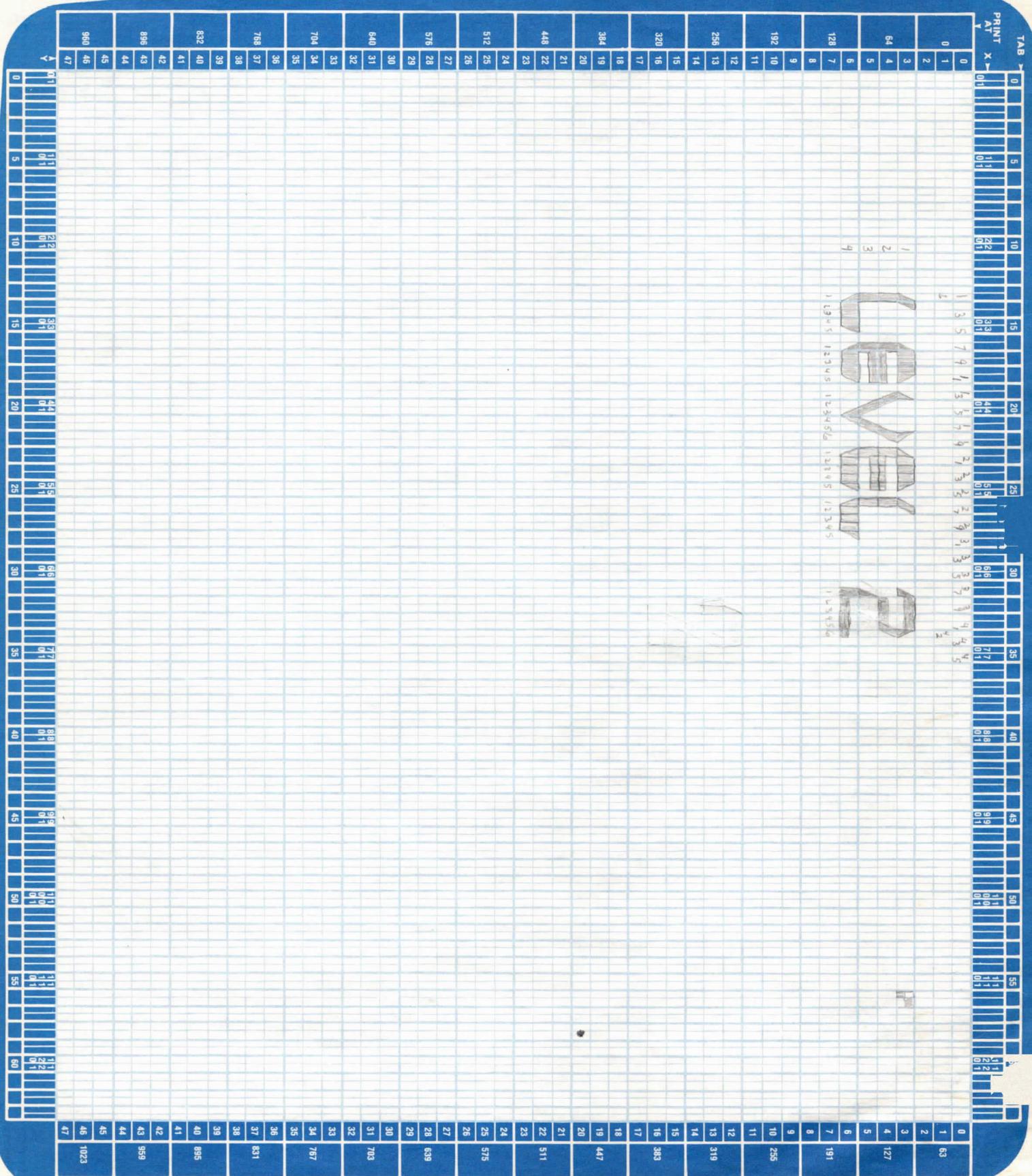


PRESS RETURN

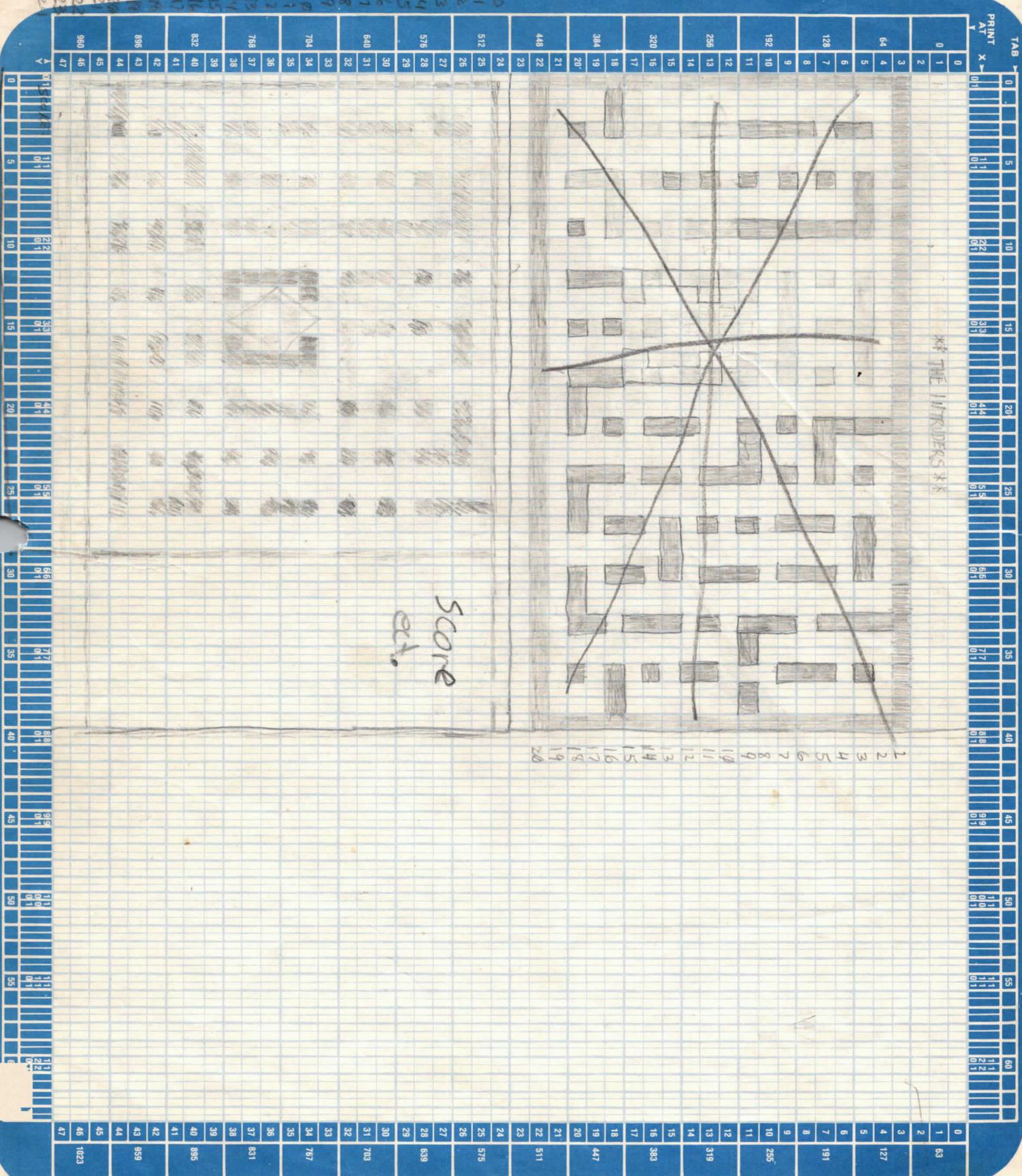
## TRS-80 Video Display Worksheet

**RadioShack®**

**TITLE** \_\_\_\_\_ **PROGRAMMER** \_\_\_\_\_ **PAGE** \_\_\_\_ OF \_\_\_\_



TITLE \_\_\_\_\_ PROGRAMMER \_\_\_\_\_ PAGE \_\_\_\_\_ OF \_\_\_\_\_



COMMENTS

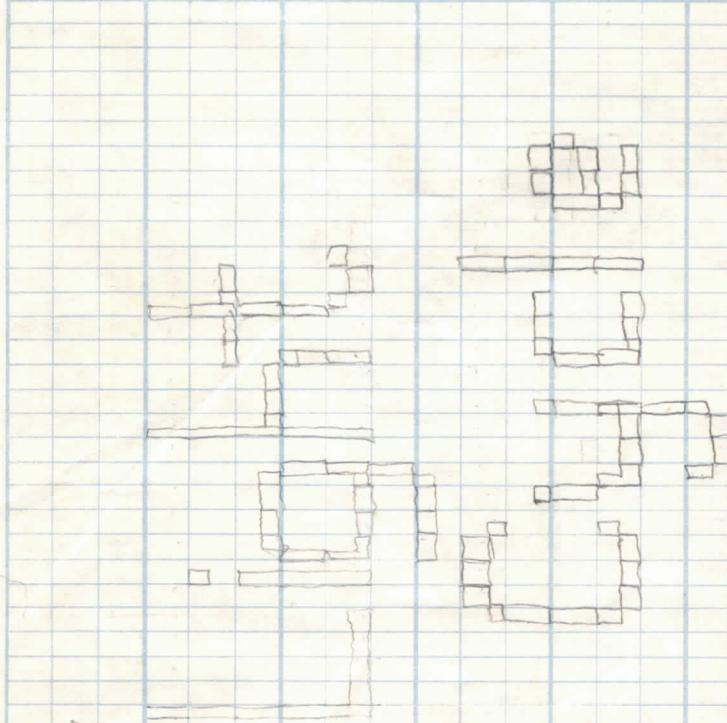
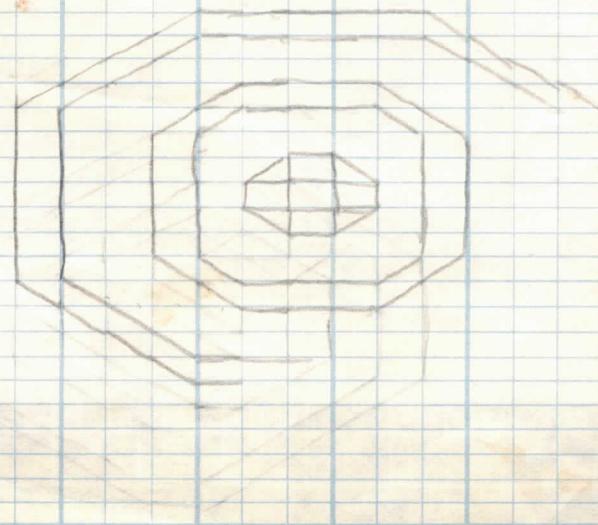
PROGRAM TITLE \_\_\_\_\_

COMMENTS

PAGE OF

PAGE OF

卷之三



## TRS-80 Video Display Worksheet

**RadioShack®**

TITLE \_\_\_\_\_ PROGRAMMER \_\_\_\_\_ PAGE \_\_\_ OF \_\_\_

# PROGRAMMER

PAGE \_\_\_\_ OF \_\_\_\_

TAB		0	5	10	15	20	25	30	35	40	45	50	55	60	65
PRINT	x	0	1	2	3	4	5	6	7	8	9	10	11	12	13
AT		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Y		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Z		0	1	2	3	4	5	6	7	8	9	10	11	12	13

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

# Radio shack

## BASIC Coding Form

**TRS-80**

TITLE	PROGRAMMER	COMMENTS	PAGE <u>OF</u>
1	Cyr		
2	User's Score;		
3			
4	High Score: 0000		
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22	Press: f1 for help f2 to change speed blue to quit		
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131			
132			
133			
134			
135			
136			
137			
138			
139			
140			
141			
142			
143			
144			
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			
165			
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			
176			
177			
178			
179			
180			
181			
182			
183			
184			
185			
186			
187			
188			
189			
190			
191			
192			
193			
194			
195			
196			
197			
198			
199			
200			
201			
202			
203			
204			
205			
206			
207			
208			
209			
210			
211			
212			
213			
214			
215			
216			
217			
218			
219			
220			
221			
222			
223			
224			
225			
226			
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237			
238			
239			
240			
241			
242			
243			
244			
245			
246			
247			
248			
249			
250			
251			
252			
253			
254			
255			
256			
257			
258			
259			
260			
261			
262			
263			
264			
265			
266			
267			
268			
269			
270			
271			
272			
273			
274			
275			
276			
277			
278			
279			
280			
281			
282			
283			
284			
285			
286			
287			
288			
289			
290			
291			
292			
293			
294			
295			
296			
297			
298			
299			
300			
301			
302			
303			
304			
305			
306			
307			
308			
309			
310			
311			
312			
313			
314			
315			
316			
317			
318			
319			
320			
321			
322			
323			
324			
325			
326			
327			
328			
329			
330			
331			
332			
333			
334			
335			
336			
337			
338			
339			
340			
341			
342			
343			
344			
345			
346			
347			
348			
349			
350			
351			
352			
353			
354			
355			
356			
357			
358			
359			
360			
361			
362			
363			
364			
365			
366			
367			
368			
369			
370			
371			
372			
373			
374			
375			
376			
377			
378			
379			
380			
381			
382			
383			
384			
385			
386			
387			
388			
389			
390			
391			
392			
393			
394			
395			
396			
397			
398			
399			
400			
401			
402			
403			
404			
405			
406			
407			
408			
409			
410			
411			
412			
413			
414			
415			
416			
417			
418			
419			
420			
421			
422			
423			
424			
425			
426			
427			
428			
429			
430			
431			
432			
433			
434			
435			
436			
437			
438			
439			
440			
441			
442			
443			
444			
445			
446			
447			
448			
449			
450			
451			
452			
453			
454			
455			
456			
457			
458			
459		</td	

## TRS-80 Video Display Worksheet

**RadioShack**

**TITLE** \_\_\_\_\_ **PROGRAMMER** \_\_\_\_\_ **PAGE** \_\_\_\_ OF \_\_\_\_

# TRS-80

## BASIC Coding Form

RadioShack

TITLE	LINE NO.	PROGRAMMER	PAGE OF	
			COMMENTS	VARIABLE LIST / COMMENTS
1	1			65 70 72 74 76 78 80
2	2			66 68 70 72 74 76 78 80
3	3			
4	4			
5	5			
6	6			
7	7			
8	8			
9	9			
10	10			
11	11			
12	12			
13	13			
14	14			
15	15			
16	16			
17	17			
18	18			
19	19			
20	20			
21	21			
22	22			
23	23			
24	24			
25	25			

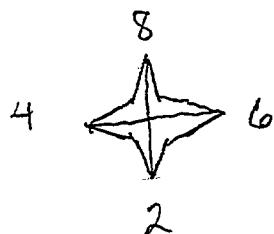
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]

You made the ALL TIME HIGH SCORE CHART  
ENTER your name, 14 character limit.

1.-----

Bonus Cycle every 15,000

controls



5 speed control

<SPACE> speed control

<P> to pause game

<Q> to quit

<W> wait

<F1> to play CYCLE MANIA

<F2> to see all time high's

<F3> to read short instructions

<F4> to select ~~score~~ difficulty

<BLUE> to quit

Difficulty increases with ~~score~~ level.

12; 40:  
24; 12

12; 40  
24; 12

$$\begin{array}{r} 30 \\ -30 \\ \hline 0 \end{array}$$

$$\begin{array}{r} (3, 0) \\ -1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} A \\ 0 \\ + \\ \hline A \end{array}$$

$$Y \begin{bmatrix} E \leftarrow 2CH(44) \\ \end{bmatrix}$$

$$\begin{array}{l} A \leftarrow 32H - 30H \\ A \leftarrow [2] + 10 \\ A \leftarrow [2] + 32H \end{array}$$

$$\begin{array}{r} 31 \\ 30 \\ \hline 10 \end{array}$$

$$\begin{array}{r} E \\ E \\ E \\ E \\ E \\ \hline \end{array}$$

$$\begin{array}{r} D \\ D \\ D \\ D \\ D \\ \hline \end{array}$$

$$\begin{array}{r} A \\ A \\ A \\ A \\ A \\ \hline \end{array}$$

$$\begin{array}{r} H \\ H \\ H \\ H \\ H \\ \hline \end{array}$$

$$\begin{array}{r} 34H \\ 36H \\ \hline 04H \end{array}$$

$$\begin{array}{r} D \\ D \\ D \\ D \\ D \\ \hline \end{array}$$

$$\begin{array}{r} A \\ A \\ A \\ A \\ A \\ \hline \end{array}$$

$$\begin{array}{r} H \\ H \\ H \\ H \\ H \\ \hline \end{array}$$

$$\begin{array}{r} D \\ D \\ D \\ D \\ D \\ \hline \end{array}$$

$$\begin{array}{r} A \\ A \\ A \\ A \\ A \\ \hline \end{array}$$

$$\begin{array}{r} H \\ H \\ H \\ H \\ H \\ \hline \end{array}$$

$$\begin{array}{r} A \\ A \\ A \\ A \\ A \\ \hline \end{array}$$

$$\begin{array}{r} D \\ D \\ D \\ D \\ D \\ \hline \end{array}$$

$$\begin{array}{r} A \\ A \\ A \\ A \\ A \\ \hline \end{array}$$

$$\begin{array}{r} D \\ D \\ D \\ D \\ D \\ \hline \end{array}$$

$$(D) = 282CH$$

$$(HL) = 282CH$$

```

1: ****
2: * H19 LIGHT CYCLE *
3: *
4: * LAST UPDATE: February 11, 1983 *
5: * PROGRAMMER: Les Bird *
6: * LANGUAGE: 8080 Assembly Language*
7: * COMPANY: CompTec software *
8: * OPERATING SYSTEM: CP/M ver 2.2 *
9: ****
10:
11: H19 escape codes
12:
13: esc equ 1bh ; escape
14: hcuh equ 'H' ; home cursor
15: hcuf equ 'C' ; cursor right
16: hcub equ 'D' ; cursor left
17: hcud equ 'B' ; cursor down
18: hcuu equ 'A' ; cursor up
19: hri equ 'I' ; reverse index
20: hcpr equ 'n' ; cursor position report
21: hsdp equ 'J' ; save cursor position
22: hrsp equ 'k' ; return to saved
23: hdca equ 'Y' ; direct addressing
24: hcd equ 'E' ; clear screen
25: hbd equ 'b' ; erase beginning
26: heop equ 'J' ; end of page
27: hel equ 'I' ; entire line
28: hebl equ 'O' ; beginning line
29: heol equ 'K' ; end of line
30: hil equ 'L' ; insert line
31: hdl equ 'M' ; delete line
32: hdch equ 'N' ; delete character
33: heim equ 'B' ; enter insert mode
34: herm equ 'O' ; exit insert mode
35: hram equ 'Z' ; reset terminal
36: hsm equ 'X' ; set mode
37: hrm equ 'Y' ; reset mode
38: herv equ 'P' ; reverse video
39: hxrv equ 'Q' ; exit reverse video
40: hegmm equ 'F' ; enter graphics mode
41: hxgm equ 'G' ; exit graphics mode
42: hf1 equ 'S' ; f1
43: hf2 equ 'T' ; f2
44: hf3 equ 'U' ; f3
45: hf4 equ 'V' ; f4
46: hf5 equ 'W' ; f5
47: hf7 equ 'P' ; blue
48: hf8 equ 'Q' ; red
49: hf9 equ 'R' ; white
50: offset equ 1fh
51: ;
52: ; bdos entry points
53: ;
54: reboot equ 0000 ; reboot system
55: bdos equ 5
56: orgin equ 100h
57: direct equ 6
58: dinput equ 0ffh
59: pstring equ 9
60: clock equ 0bh

```

```
61: cr      equ     0dh
62: lf      equ     0ah
63: bs      equ     8
64: ap      equ     27h
65: bel     equ     7
66: ; file equates
67: dma     equ     0080h ; direct memory access
68: fcb     equ     005ch ; file control block
69: openf   equ     0fh    ; open file
70: closef  equ     10h    ; close file
71: delf    equ     13h    ; delete file
72: readf   equ     14h    ; read file
73: writef  equ     15h    ; write file
74: makef   equ     16h    ; make file
75: ;
76:         org     origin
77: start: lxi    h,0
78:         dad    sp
79:         shld   oldstack
80:         lxi    sp,stack
81:         call   highfile
82:         jmp    begin
83: ; CLEAR DISPLAY
84: cls:    push   h
85:         lxi    h,101h
86:         shld   curco
87:         call   adj    ; incase on 25th line
88:         mvi   a,hcd
89:         call   process
90:         lxi    h,memmap
91:         shld   scrpnt
92:         pop    h
93: zsmem:  push   h
94:         lxi    h,memmap
95:         lxi    d,1920
96: zsmemi: mvi   m,32
97:         inx    h
98:         dex    d
99:         mov    a,e
100:        ora   d
101:        jnz   zsmemi
102:        pop    h
103:        ret
104: ;
105: ceol:   push   h
106:         mvi   a,heol
107:         call  process
108:         lhld   curco
109:         xchg
110:         lhld   scrpnt
111: eolloop:mvi  m,0
112:         inx    h
113:         dcr    d
114:         jnz   eolloop
115:         pop    h
116:         ret
117: crlf:   mvi   a,cr
118:         call  output
119:         mvi   a,lf
120:         call  output
```

```
121:      push    h
122:      lhld    curco
123:      mvi    h,1
124:      inn    l
125:      shld    curco
126:      pop    h
127:      ret
128: ;      DELETE CHARACTER routine
129: deletec:
130:      mvi    a,hdch
131:      jmp    process
132: ;      CURSOR HOME routine
133: home:
134:      push   h
135:      mvi    a,hcuh
136:      call   process
137:      lxi    h,101h
138:      shld   curco
139:      lxi    h,memmap
140:      shld   scrpnt
141:      pop    h
142:      ret
143: ;      PROCESS ESCAPE SEQUENCE
144: process:
145:      push   psw
146:      mvi    a,esc
147:      call   output
148:      pop    psw
149:      jmp    output
150: hset:
151:      push   psw
152:      mvi    a,hsm
153: hsetm:
154:      call   process
155:      pop    psw
156:      call   output
157:      ret
158: ;      CLEAR LINE 25
159: clr25:
160:      mvi    a,'1'
161:      call   hrset
162:      mvi    a,'1'
163:      jmp    hset
164: ;      OUTPUT A SINGLE CHARACTER
165: output:
166:      mov    e,a
167:      mvi    c,direct
168:      jmp    bentry
169: ;      CHECK FOR INPUT IN BUFFER
170: input:
171:      mvi    e,dinput
172:      mvi    c,direct
173:      jmp    bentry
174: ;      WAIT FOR INPUT IN BUFFER
175: wait:
176:      lxi    h,0ffffh; delay time
177: wait1:
178:      push   h
179:      call   input
180:      pop    h
181:      ora    a
182:      rnz
183:      dex    h
184:      mov    a,h
```

```
181:         ora     l
182:         jnz     wait1
183:         ret
184: ;      BDOS ENTRY
185: bentry: push    b
186:         push    d
187:         push    h
188:         call    bdos
189:         pop     h
190:         pop     d
191:         pop     b
192:         ret
193: ;      SAVE ALL REGISTERS ON STACK
194: savall: xthl
195:         push    d
196:         push    b
197:         push    psw
198:         push    h
199:         ret
200: ;      RETRIEVE ALL REGISTERS ON STACK
201: retall: pop     h
202:         pop     psw
203:         pop     b
204:         pop     d
205:         xthl
206:         ret
207: ;      ADJUST CURSOR POSITION (H=X, L=Y)
208: adj:   push    h
209:         mvi    a,hdca
210:         call   process
211:         mov    a,l
212:         adi    offset
213:         call   output
214:         mov    a,h
215:         adi    offset
216:         call   output
217:         pop    h
218: adjmem: push   h      ; ADJUST MEMORY MAP ONLY
219:         mov    b,h
220:         mov    c,l
221:         lxi   h,memmap
222:         lxi   d,50h
223:         dcr   c
224:         jz    adjloop1
225: adjloop: dad
226:         dcr   c
227:         jnz   adjloop
228: adjloop1:
229:         mov    e,b
230:         mvi   d,0
231:         dad   d
232:         shld  scrpnt
233:         pop    h
234:         ret
235: ;      UPDATE MEMORY MAP
236: update: push   h
237:         lhld  scrpnt
238:         mov    m,a
239:         inx   h
240:         shld  scrpnt
```

```
241:          pop      h
242:          ret
243: ;       READ MEMORY MAP
244: rsmem:   push     h
245:          call     adjmem ; adjust memory pointer
246:          lhld    scrpnt
247:          mov     a,m
248:          pop     h
249:          ret
250: ;       STORE CHARACTER IN MEMORY MAP
251: ssmem:   push     h
252:          call     adjmem ;
253:          lhld    scrpnt
254:          mov     m,a
255:          pop     h
256:          ret
257: ;       ENTER GRAPHICS MODE
258: graphix:
259:          mvi    a, heg'm
260:          call   process
261:          mvi    a,1
262:          sta    gbit
263:          ret
264: ;       EXIT GRAPHICS MODE
265: xgraphix:
266:          mvi    a, hxgm
267:          call   process
268:          mvi    a,0
269:          sta    gbit
270:          ret
271: ;       ENTER REVERSE VIDEO MODE
272: reverse:
273:          mvi    a, herv
274:          call   process
275:          mvi    a,1
276:          sta    rbit
277:          ret
278: ;       EXIT REVERSE VIDEO MODE
279: xreverse:
280:          mvi    a, hxrv
281:          call   process
282:          mvi    a,0
283:          sta    rbit
284:          ret
285: ;       TURN CURSOR OFF
286: cursoff:
287:          mvi    a, 'x'
288:          call   process
289:          mvi    a, '5'
290:          jmp    output
291: ;       TURN CURSOR ON
292: cursor:
293:          mvi    a, 'y'
294:          call   process
295:          mvi    a, '5'
296:          jmp    output
297: ;       MAKE CURSOR UNDERLINE
298: cursl:   mvi    a, hrm
299:          call   process
300:          mvi    a, '4'
```

```
301:         jmp      output
302: ;      MAKE CURSOR BLOCK
303: cursbs: mvi      a,hsm
304:         call     process
305:         mvi      a,'4'
306:         jmp      output
307: ;      SAVE CURSOR POSITION
308: savecurs:
309:         mvi      a,hscp
310:         jmp      process
311: ;      RETRIEVE CURSOR TO SAVED POSITION
312: retcurs:
313:         mvi      a,hrcp
314:         jmp      process
315: ;      EXIT TO CP/M
316: boot:
317: restore:
318:         call    wrhigh ; write high scores
319:         call    closefile
320:         mvi    a,hram
321:         call    process
322:         lhld   oldstack
323:         sphl   ; set stack to oldstack
324:         ret    ; return to CCP
325: ;      PRINT STRING ON CRT
326: show:   mov    a,m
327:         cpi    '@'
328:         jz     printat
329:         cpi    '['
330:         jz     special
331:         cpi    0
332:         rz
333:         inx    h
334:         call   update
335:         call   output
336:         jmp    show
337: printat:
338:         inx    h
339:         mov    c,m
340:         inx    h
341:         mov    b,m
342:         inx    h
343:         call   savall
344:         mov    l,c
345:         mov    h,b
346:         call   adj
347:         call   retall
348:         jmp    show
349: special:
350:         inx    h
351:         mov    a,m
352:         cpi    'R'
353:         cz    reverse
354:         cpi    'r'
355:         cz    xreverse
356:         cpi    'G'
357:         cz    graphix
358:         cpi    'g'
359:         cz    xgraphix
360:         cpi    'S'
```

```
361:     cz      setmodes
362:     cpi    's'
363:     cz      rsetmodes
364:     cpi    'C'
365:     cz      cls
366:     cpi    'c'
367:     cz      ceol
368:     cpi    'J'
369:     jnz    special
370:     inx    h
371:     jmp    show
372: setmodes:
373:     mvi    a,hsm
374: setentry:
375:     call   process
376:     inx    h
377:     mov    a,m
378:     jmp    output
379: rsetmodes:
380:     mvi    a,hrm
381:     jmp    setentry
382: delay:  call  savall
383:     lhld   time
384: delay1: dex   h
385:     mov    a,h
386:     ora    l
387:     jnz    delay1
388:     call   retall
389:     ret
390: ;       INPUT A LINE FROM THE KEYBOARD
391: inbuf:  push   h
392:     call   wait
393:     pop    h
394:     cpi    cr
395:     rz
396:     cpi    bs
397:     jz     inbuf2
398:     cpi    3      ; ctrl-c
399:     jz     xinbuf
400:     cpi    20h
401:     jc     inbuf
402:     cpi    61h
403:     jc     inbuf1
404:     ani    5fh
405: inbuf1: mov    m,a
406:     push   h
407:     call   output
408:     pop    h
409:     inx    h
410:     jmp    inbuf
411: inbuf2: dex   h
412:     mov    a,m
413:     cpi    i
414:     jz     inbuf3
415:     push   h
416:     mvi    a,bs
417:     call   output
418:     mvi    a,'_'
419:     call   output
420:     mvi    a,bs
```

```
421:         call    output
422:         pop     h
423:         jmp    inbuf
424: inbuf3: mvi    a,7      ; bell
425:         push   h
426:         call   output
427:         pop    h
428:         inx    h
429:         jmp    inbuf   ; loop
430: xinbuf: pop    b
431:         jmp    begin
432: ; sets up File Control Block for high scores
433: fcbset: lda    drive
434:         sta    fcb    ; drive #
435:         lxi    h,filename
436:         lxi    d,fcb+1
437: fcbset1:mov    a,m
438:         ora    a
439:         jz    fcbset2 ; write .TYP
440:         stax   d
441:         inx    d
442:         inx    h
443:         jmp    fcbset1
444: fcbset2:lxi    h,filetype
445: fcbset3:mov    a,m
446:         ora    a
447:         jz    fcbset4
448:         stax   d
449:         inx    h
450:         inx    d
451:         jmp    fcbset3
452: fcbset4:lda    extnum
453:         sta    fcb+12
454:         lda    recnt
455:         sta    fcb+15
456:         lda    currrec
457:         sta    fcb+32
458:         lhld   recnum
459:         shld   fcb+33
460:         xra    a
461:         sta    fcb+35
462:         ret
463: ; open the HIGHSCORE file on the disk
464: openfile:
465:         call   fcbset
466:         mvi    c,openf
467:         lxi    d,fcb
468:         call   bdos
469:         cpi    0ffh    ; no file to open
470:         rnz
471: makefile:
472:         call   fcbset
473:         mvi    c,makef
474:         lxi    d,fcb
475:         call   bdos
476:         cpi    0ffh
477:         jz    nserror ; no space
478:         ret
479: ; close file
480: closefile:
```

```

481:     mvi      c,closef
482:     lxi      d,fcb
483:     call     bdos
484:     cpi     @ffh
485:     jz      nferror
486:     ret
487: ; read data from file
488: readfile:
489:     mvi      c,readf
490:     lxi      d,fcb
491:     call     bdos
492:     ret
493: ; write data to file. @ffh=end of file
494: writefile:
495:     mvi      c,writef
496:     lxi      d,fcb
497:     call     bdos
498:     ret
499: ; delete file
500: delfile:mvi   c,delf
501:     lxi      d,fcb
502:     call     bdos
503:     ret
504: ;
505: ;          LISTING PROGRAM:          CYCLE19.ASM
506: ;          LISTING DATE:           March 15, 1983
507: ;
508: ;
509: ; USER PROGRAM STARTS HERE
510: ;
511: ; contains programmable delay - HL=delay, SHLD TIME
512: count1: ds    2      ; 32 counts
513: begin: lxi   h,0020h
514:     shld   count1
515:     lxi   h,intro
516:     call   show   ; print heading
517:     call   cursoff
518: begin1: call  diskill ; display skill level
519:     call  copyright
520:     lhld  count1
521:     dcx   h
522:     mov   a,h
523:     ora   l
524:     jz   cycle19 ;
525:     shld  count1
526:     mvi   a,'i'
527:     sta   boxchar
528:     call  box1
529:     call  begin3
530:     mvi   a,' '
531:     sta   boxchar
532:     call  box7
533:     call  begin3
534:     call  copyright
535:     call  box1
536:     call  begin3
537:     mvi   a,'i'
538:     sta   boxchar
539:     call  box7
540:     call  begin3

```

```
541:         jmp      begin1
542: begin3: call     input
543:         ora      a
544:         jnz      begin2
545:         ret
546: begin2: pop     b      ; return address
547:         lxi     h,0020h
548:         shld    count1
549:         cpi     esc
550:         jz      escvec
551:         cpi     'q'
552:         jz      boot
553:         cpi     'w'
554:         jz      cycle19
555:         jmp      begin1
556: escvec: call    wait
557:         cpi     hfi
558:         jz      pgame
559:         cpi     hf2
560:         jz      highsore
561:         cpi     hf3
562:         jz      help
563:         cpi     hf4
564:         jz      selspd
565:         cpi     hf5
566:         jz      selllev ; select level
567:         cpi     hf7
568:         jz      boot
569:         jmp      begin1
570: selspd: lxi    h,mess10
571:         call    show
572:         call    wait
573:         call    wait
574:         cpi     hfi
575:         jz      fast
576:         cpi     hf2
577:         jz      slow1
578:         cpi     hf3
579:         jz      slow2
580:         cpi     hf4
581:         jz      slow3
582:         jmp      selspd
583: fast:   lxi    h,0100h
584:         shld    time
585:         mvi     a,1
586:         sta      skill
587:         sta      lev2
588:         jmp      begin
589: slow1:  lxi    h,0200h
590:         shld    time
591:         mvi     a,2
592:         sta      skill
593:         jmp      begin
594: slow2:  lxi    h,0700h
595:         shld    time
596:         mvi     a,3
597:         sta      skill
598:         jmp      begin
599: slow3:  lxi    h,1200h
600:         shld    time
```

```
601:         mvi    a,4
602:         sta    skill
603:         jmp    begin
604: ;
605: sellev: call   dislevel1
606:         lxi    h,levmes1
607:         call   show
608: selllevi:call  wait
609:         cpi   '9'+1 ; highest level to start
610:         jnc   sellvi ; higher than nine
611:         cpi   '1'
612:         jc    sellvi ; less than zero
613:         sui   31h ; make binary number
614:         sta    lev2 ; adjust level
615:         call   graphix ; enter graphics mode
616:         call   dislevel2
617:         jmp    begin
618: ;
619: diskill:lda   skill ; get skill level
620:         cpi   1 ; expert skill
621:         jz    skill1
622:         cpi   2 ; pro skill
623:         jz    skill2
624:         cpi   3 ; intermediate skill
625:         jz    skill3
626:         cpi   4 ; amateur skill
627:         jz    skill4
628: skill1:lxi   h,expmes
629:         call  show
630:         ret
631: skill2:lxi   h,promes
632:         call  show
633:         ret
634: skill3:lxi   h,intmes
635:         call  show
636:         ret
637: skill4:lxi   h,amames
638:         call  show
639:         ret
640: pgame:mvi   a,4
641:         sta    cycleft
642:         lxi    h,3030h
643:         shld  score1
644:         shld  score1+2
645:         mvi   a,'0'
646:         sta    score1+4
647:         mvi   a,i
648:         sta    lev
649:         sta    lev1
650:         sta    level
651:         sta    cdbit2
652:         sta    cdbit3
653:         mvi   a,0
654:         sta    cdbit1
655:         sta    lev3
656:         call  cursoff
657:         lxi    h,freebie
658:         shld  freepnt
659: game:call   graphix ; enter graphics mode
660:         call  dislevel; display level on screen
```

```
661:           call    xgraphix; exit
662:           lxi    h, grid
663:           call    show
664:           call    init
665:           call    place
666:           call    misc
667: loop1:  call    wait
668:           cpi    'q'
669:           jz     begin
670:           sta    plmove
671: game1:   call    delay
672:           call    player
673:           lda    plabit
674:           ora    a
675:           jz     game
676:           call    computr
677:           lda    combit
678:           ora    a
679:           jz     game
680:           call    delay
681:           jmp    game1
682: place:   lhld   playco
683:           call    adj
684:           lda    pldir
685:           call    grout
686:           lhld   comco
687:           call    adj
688:           lda    comdir
689:           call    grout
690:           lda    levi
691:           cpi    2
692:           rc
693:           jnz    place2
694: place1:  lhld   comco2
695:           call    adj
696:           lda    comdir
697:           call    grout
698:           ret
699: place2:  call    place1
700:           lhld   comco3
701:           call    adj
702:           lda    comdir
703:           call    grout
704:           ret
705: player:  lhld   placo
706:           shld   oldco
707:           call    input
708:           ora    a
709:           jz     nomove
710:           cpi    '5'
711:           jz     pspeed
712:           cpi    ' '
713:           jz     pspeed
714:           cpi    'p'
715:           jz     pause
716:           cpi    'p'
717:           jz     pause
718:           cpi    'q'
719:           jz     begin
720:           sta    plmove
```

```
721:          cpi    '4'
722:          jz     pleft
723:          cpi    '6'
724:          jz     pright
725:          cpi    '2'
726:          jz     pdown
727: pup:      dcr    1
728:          shld   plac0
729:          mov    a,1
730:          cpi    3
731:          jz     pldead
732:          mvi   a,'u'
733:          sta    pldir
734:          jmp    xplayer
735: pdowns:   inr    1
736:          shld   plac0
737:          mov    a,1
738:          cpi    23
739:          jz     pldead
740:          mvi   a,'s'
741:          sta    pldir
742:          jmp    xplayer
743: pleft:    dcr    h
744:          shld   plac0
745:          mov    a,h
746:          cpi    1
747:          jz     pldead
748:          mvi   a,'t'
749:          sta    pldir
750:          jmp    xplayer
751: pright:   inr    h
752:          shld   plac0
753:          mov    a,h
754:          cpi    80
755:          jz     pldead
756:          mvi   a,'v'
757:          sta    pldir
758:          jmp    xplayer
759: nomove:   lda    plmove
760:          cpi    '2'
761:          jz     pdown
762:          cpi    '4'
763:          jz     pleft
764:          cpi    '6'
765:          jz     pright
766:          jmp    pup
767: pspeed:   lda    speed1
768:          cpi    1
769:          jz     pspeed1
770:          mvi   a,1
771:          sta    speed1
772:          jmp    player
773: pspeed1:mvi a,2
774:          sta    speed1
775:          jmp    player
776: xplayer:lhld playco
777:          call   adj
778:          lda    pldir
779:          call   grout
780:          call   rsmem
```

```

781:      cpi      " "
782:      jnz      pldead
783:      lhld    oldco
784:      call    adj
785: plchar: mvi    a,'i'
786:      call    grout
787:      call    ssmem
788:      call    blbon   ; block bonus
789:      lda    speed1
790:      cpi    4
791:      jz     pspeed2
792:      cpi    2
793:      jz     pspeed3
794:      ret
795: ;
796: blbon: lxi    h,blocks
797:      inx    h
798:      inx    h
799: blbon1: mov    a,m      ; get tens unit
800:      inn    a
801:      cpi    ":"      ; test for 10
802:      jz     blbon2
803:      mov    m,a
804:      ret      ; ten points added
805: blbon2: mvi    m,'0'    ; ascii zero
806:      dcx    h
807:      jm    -04h [REDACTED]; z",#do[REDACTED]J4k+[REDACTED]oo[REDACTED]/P
Fall  show
808:      lxi    h,score1; player 1 score
809:      xchg   ; put in DE
810:      lxi    h,blocks; bonus points
811:      inx    d      ; 10,000's digit
812:      inx    d      ; 1,000's digit
813:      inx    d      ; 100's digit
814:      inx    d      ; tens position
815:      inx    h
816:      inx    h      ; tens position
817: addbonus1:
818:      push   h
819:      push   d
820:      mvi    a,' '    ; delimiter
821:      cmp    m
822:      jz     addbonus4
823:      ldax   d
824:      sui    30h
825:      add    m
826:      cpi    ":"'
827:      jc     addbonus3
828: addbonus2:
829:      sui    0ah      ; subtract ten
830:      stax   d      ; put in score
831:      dcx    d      ; next unit
832:      ldax   d      ; get number
833:      inn    a      ; increment count
834:      cpi    ":"'
835:      jz     addbonus2
836: addbonus3:
837:      stax   d
838:      pop    d
839:      pop    h
840:      dex    h

```

```
841:         dcx      d
842:         jmp      addbonus1
843: addbonus4:
844:         pop      d
845:         pop      h
846:         lxi      h, 3030h ; zero block count
847:         shld    blocks
848:         shld    blocks+2
849:         call    scorout ; update score
850:         ret
851: ;
852: bonusmes:
853:         db      'B', 13, 28, 'BONUS + '
854: blocks: db      '0000 POINTS ', 0
855: ;
856: pspeed2:mvi   a, 2
857:         sta      speed1
858:         ret
859: pspeed3:mvi   a, 4
860:         sta      speed1
861:         jmp      player
862: grout: push   psw
863:         call    graphix
864:         pop      psw
865:         call    output
866:         push   psw
867:         call    xgraphix
868:         pop      psw
869:         ret
870: pldead: mvi   a, 0
871:         sta      plabit
872:         lda      cycleft
873:         dcr      a
874:         jz      gameov
875:         sta      cycleft
876:         mvi   a, 1
877:         sta      speed1
878:         lxi      h, pdmes
879:         call   show
880: reinit: lda   lev
881:         cpi   i
882:         jz      rein1
883:         cpi   2
884:         jz      rein2
885: rein3: mvi   a, 0
886:         sta      cdbit3
887:         sta      cdbit2
888:         sta      cdbit1
889:         ldi      loop2
890: rein2: mvi   a, 0
891:         sta      cdbit2
892:         sta      cdbit1
893:         mvi   a, 1
894:         sta      cdbit3
895:         jmp      loop2
896: rein1: mvi   a, 0
897:         sta      cdbit1
898:         mvi   a, 1
899:         sta      cdbit2
900:         sta      cdbit3
```

```
901: loop2: call    addbonus
902: loop2b: call    wait
903:          cpi    cr
904:          jnz    loop2b
905:          call    clr25
906:          ret
907: gameov: call    newhigh
908:          lxi    h,gomes
909:          call    show
910:          call    wait
911:          ori    20h      ; make lower case
912:          cpi    'y'
913:          jz     pgame
914:          cpi    'n'
915:          jz     begin
916:          cpi    'q'
917:          jz     restore
918:          jmp    gameov
919: gomes: db     '[[C]@',12,34,'[R] GAME OVER [r]@',15,20
920:          db     'press [R] Y [r] to play again, [R] N [r] to quit.',0
921: ;
922: cload: lda   level
923:          cpi    1
924:          jz     load1
925:          cpi    2
926:          jz     load2
927: load3: lda   cdbit3
928:          ora    a
929:          jnz    xload1
930:          lhld   comco3
931:          mvi    a,3      ; thought pattern 3
932:          sta    thought
933:          ret
934: load2: lda   cdbit2
935:          ora    a
936:          jnz    xload
937:          lhld   comco2
938:          mvi    a,1      ; thought pattern 1
939:          sta    thought
940:          ret
941: load1: lda   cdbiti
942:          ora    a
943:          jnz    xload
944:          lhld   comco1
945:          mvi    a,2      ; thought pattern
946:          sta    thought
947:          ret
948: xload: lda   level
949:          inr    a
950:          mov    b,a
951:          mvi    a,3
952:          cmp    b
953:          jnc    load4
954:          mvi    b,1
955: load4: mov    a,b
956:          sta    level
957:          jmp    cload
958: xload1: mvi    a,1
959:          sta    level
960:          mvi    a,0ffh
```

```
961:         ret
962: cstор:    lda     level
963:           cpi     1
964:           jz      stor1
965:           cpi     2
966:           jz      stor2
967: stor3:    shld   comco3
968:           ret
969: stor2:    shld   comco2
970:           ret
971: stor1:    shld   comcol
972:           ret
973: computr: call  cload
974:           cpi   0ffh
975:           rz
976:           jmp   cload1
977: computri:
978:           mov   a,e
979:           cmp   l
980:           jz    compx
981:           jc    comup
982:           jmp   comdwn
983: computr2:           ; thought pattern 2
984:           mov   a,d
985:           cmp   h
986:           jz    computri
987:           jc    comlft
988:           jmp   comrgt
989: computr3:           ; thought pattern 3
990:           mov   a,e
991:           cmp   l
992:           jz    computr2
993:           jc    computr4
994:           mov   a,e
995:           sui   4           ; four space miss
996:           cmp   l           ; test against player
997:           jc    comup
998:           jmp   comdwn
999: computr4:           ; thought pattern 3b
1000:          mov   a,e           ; py
1001:          adi   4
1002:          cmp   l           ; (py+4)-cy
1003:          jc    comup
1004:          jmp   comdwn
1005: ;
1006: compx:   mov   a,d
1007:           cmp   h
1008:           jc    comlft
1009: comrgt:  call  cload
1010:           irr   h
1011:           call  rsmem
1012:           cpi   ''
1013:           jnz   comlft
1014:           call  cstор
1015:           mvi   a,'v'
1016:           sta   combit
1017:           jmp   xcomp
1018: comlft:  call  cload
1019:           dcr   h
1020:           call  rsmem
```

```
1021:      cpi      ? ?
1022:      jnz      comup
1023:      call    ctor
1024:      mvi    a, 't'
1025:      sta    combit
1026:      jmp    xcomp
1027: comup:  call    cload
1028:          dcr    1
1029:          call   rsmem
1030:          cpi    ?
1031:          jnz    comdwn
1032:          call   ctor
1033:          mvi    a, 'u'
1034:          sta    combit
1035:          jmp    xcomp
1036: comdwn: call    cload
1037:          inn    1
1038:          call   rsmem
1039:          cpi    ?
1040:          jnz    tryag
1041:          call   ctor
1042:          mvi    a, 's'
1043:          sta    combit
1044:          jmp    xcomp
1045: tryag:  lda    combit4
1046:          cpi    2
1047:          jz    comdead
1048:          mvi    a, 2
1049:          sta    combit4
1050:          jmp    comrgt
1051: xcomp:  call    adj    ; cursor adjustment
1052:          call   reverse ; reverse computer cyc's
1053:          lda    combit
1054:          call   grout
1055:          call   upscrn
1056:          call   xreverse
1057:          lhld   oldcol
1058:          call   adj    ; cursor adjust
1059:          call   ssmem
1060:          mvi    a, 'i'
1061:          call   grout
1062:          mvi    a, 1
1063:          sta    combit4
1064:          call   chkspd
1065:          ora    a
1066:          jnz    cspeed2
1067: reent:  lda    lev
1068:          mov    b, a
1069:          lda    level
1070:          cmp    b
1071:          jz    xcompi
1072:          inn    a
1073:          sta    level
1074:          jmp    computr
1075: cspeed2: lda    cspeed1
1076:          cpi    4
1077:          jnz    cspeed3
1078:          mvi    a, 2
1079:          sta    cspeed1
1080:          jmp    reent
```

```
1081: cspeed3:mvi    a, 4
1082:           sta    cspeed1
1083:           jmp    computr
1084: xcompi: mvi    a, 1
1085:           sta    level
1086:           ret
1087: chkspd: lda    lev1
1088:           cpi    5
1089:           jz     fast3
1090:           cpi    7
1091:           jz     fast2
1092:           cpi    9
1093:           jz     fast1
1094: lda    lev2
1095: ora    a
1096: jnz    allfast
1097: mvi    a, 0
1098: ret
1099: fast3: lda    level
1100: cpi    3
1101: mvi    a, 0
1102: rnz
1103: mvi    a, 1
1104: ret
1105: fast2: lda    level
1106: cpi    2
1107: mvi    a, 0
1108: rnz
1109: mvi    a, 1
1110: ret
1111: fast1: lda    level
1112: cpi    1
1113: mvi    a, 0
1114: rnz
1115: mvi    a, 1
1116: ret
1117: allfast:mvi  a, 1
1118: ret
1119: comdead:mvi  a, 7
1120: call   output
1121: call   incscor
1122: call   comera
1123: call   setbit
1124: call   comtst
1125: cpi   0ffh
1126: jnz   computr
1127: call   cdmount ; pick random sentence
1128: call   show
1129: comded1:lda  lev1
1130: inn   a
1131: sta    lev1
1132: push   psw
1133: lda    lev1
1134: cpi    10
1135: jnz   comded2
1136: lda    lev2
1137: inn   a
1138: sta    lev2
1139: cpi    10
1140: jnz   comded3
```

1141: lda lev3  
1142: inr a  
1143: sta lev3  
1144: xra a ; zero second digit  
1145: sta lev2  
1146: comded3:xra a  
1147: sta levi  
1148: comded2:pop psw  
1149: cpi 3  
1150: jc levinc  
1151: mvi a,3  
1152: levinc: sta lev  
1153: cpi i  
1154: jz aliv1  
1155: cpi 2  
1156: jz aliv2  
1157: aliv3: mvi a,0  
1158: sta cdbit3  
1159: aliv2: mvi a,0  
1160: sta cdbit2  
1161: aliv1: mvi a,0  
1162: sta cdbit1  
1163: loop3: call addbonus  
1164: loop3b: call wait  
1165: cpi cr  
1166: jnz loop3b  
1167: mvi a,1  
1168: sta speedi  
1169: call clr25  
1170: mvi a,0  
1171: sta combit  
1172: mvi a,1  
1173: sta combit4  
1174: ret  
1175: setbit: lda level  
1176: cpi 1  
1177: jz setbit1  
1178: cpi 2  
1179: jz setbit2  
1180: setbit3:mvi a,1  
1181: sta cdbit3  
1182: ret  
1183: setbit2:mvi a,1  
1184: sta cdbit2  
1185: ret  
1186: setbit1:mvi a,1  
1187: sta cdbit1  
1188: ret  
1189: comtst: lda level  
1190: cpi 1  
1191: jz tst1  
1192: cpi 2  
1193: jz tst2  
1194: tst3: lda cdbit3  
1195: ora a  
1196: rz  
1197: jmp xtst  
1198: tst2: lda cdbit2  
1199: ora a  
1200: rz

```
1201:          jmp      xtst
1202: tsti:      lda      cdbit1
1203:          ora      a
1204:          rz
1205: xtst:      lda      cdbit1
1206:          ora      a
1207:          rz
1208:          lda      cdbit2
1209:          ora      a
1210:          rz
1211:          lda      cdbit3
1212:          ora      a
1213:          rz
1214:          mvi      a, 0ffh
1215:          ret
1216: upscrn:    lda      level
1217:          cpi      1
1218:          jz       upscr1
1219:          cpi      2
1220:          jz       upscr2
1221: upscr3:    lhld    comco3
1222:          xchg
1223:          lhld    screen3
1224:          mov      m, e
1225:          inx      h
1226:          mov      m, d
1227:          inx      h
1228:          shld    screen3
1229:          mvi      m, 0
1230:          ret
1231: upscr2:    lhld    comco2
1232:          xchg
1233:          lhld    screen2
1234:          mov      m, e
1235:          inx      h
1236:          mov      m, d
1237:          inx      h
1238:          shld    screen2
1239:          mvi      m, 0
1240:          ret
1241: upscr1:    lhld    comcol
1242:          xchg
1243:          lhld    screen1
1244:          mov      m, e
1245:          inx      h
1246:          mov      m, d
1247:          inx      h
1248:          shld    screen1
1249:          mvi      m, 0
1250:          ret
1251: comera:   lda      level
1252:          cpi      1
1253:          jz       erac1
1254:          cpi      2
1255:          jz       erac2
1256: erac3:     lxi      h, memmap3
1257: erac3a:    mov      a, m
1258:          ora      a
1259:          rz
1260:          mov      e, a
```

```

1261:     inx    h
1262:     mov    a, m
1263:     ora    a
1264:     rz
1265:     mov    d, a
1266:     inx    h
1267:     push   h
1268:     mov    h, d
1269:     mov    l, e
1270:     call   adj
1271:     mvi   a, ' '
1272:     call   update
1273:     call   output
1274:     pop    h
1275:     jmp   erac3a
1276: erac2: lxi   h, memmap2
1277:     jmp   erac3a
1278: erac1: lxi   h, memmap1
1279:     jmp   erac3a
1280: cdmout: lxi   h, retmes; return message
1281:     call   show
1282:     lhld  clock ; get clock counter
1283:     mov   a, m ; get counter
1284:     ral   ; rotate left
1285:     jc    cdmout1 ; 1st message
1286:     ral   ; rotate left again
1287:     ral   ; rotate three times
1288:     jc    cdmout2 ; 2nd message
1289: cdmout3: lxi   h, cdmes3; 3rd message
1290:     ret
1291: cdmout2: lxi   h, cdmes2; 2nd message
1292:     ret
1293: cdmout1: lxi   h, cdmes1; 1st message
1294:     ret
1295: cdmes1: db    ' [S1]@', 25, 1, ' [Rc]Users think there so good. [r]', 0
1296: cdmes2: db    ' [S1]@', 25, 1, ' You got lucky!! If you think your'
1297:         db    ' so good, increase the level.', 0
1298: cdmes3: db    ' [S1]@', 25, 1, ' Alright, you''re getting me angry.'
1299:         db    ' I''ll have to send my best warriors.', 0
1300: pdmes: db    ' [S1]@', 25, 1, ' I will take over the world!!!', 0
1301: retmes: db    '@', 24, 1, ' [c]@', 24, 34, 'PRESS [R] RETURN [r]', 0
1302: ;
1303: cload1: shld  oldco1 ; save for wall
1304:         xchg  ; computer x,y
1305:         lhld  playco ; player x,y
1306:         xchg  ; DE=px, py HL=cx, cy
1307:         push   h
1308:         push   d
1309:         lda   lev3 ; high level bit
1310:         ora   a
1311:         jz    cload2 ; high bit = 0
1312:         adi   09h ; add 9
1313: cload2: mov   b, a ; store in B
1314:         lda   lev2 ; low level bit
1315:         add   b ; add reg. B
1316:         cpi   28 ; highest level
1317:         jc    cload3 ; subtract 27
1318:         sui   ibh ; 
1319: cload3: mov   b, a ; store in B
1320:         mvi   c, 6 ; multiply by 7

```

```

1321:      xra      a      ; clear accumalator
1322:  cload4: add      b      ;
1323:          dcr      c
1324:          jnz      cload4
1325:          mov      e,a
1326:          lda      thought
1327:          dcr      a
1328:          add      a      ; double i=2,2=4
1329:          add      e
1330:          mvi      d,0      ;
1331:          mov      e,a      ; put in E
1332:          lxi      h, patterns
1333:          dad      d
1334:          mov      c,m
1335:          inx      h      ; next location
1336:          mov      b,m
1337:          pop      d
1338:          pop      h
1339:          push     b
1340:          ret      ; return to selected move
1341: ;
1342: patterns:
1343:      dw      computri,computri1,computri      ; 1
1344:      dw      computr2,computri1,computri      ; 2
1345:      dw      computr3,computri1,computri      ; 3
1346:      dw      computri1,computri1,computr2      ; 4
1347:      dw      computr2,computri1,computr2      ; 5
1348:      dw      computr3,computri1,computr2      ; 6
1349:      dw      computri1,computri1,computr3      ; 7
1350:      dw      computr2,computri1,computr3      ; 8
1351:      dw      computr3,computri1,computr3      ; 9
1352:      dw      computri1,computr2,computri      ; 10
1353:      dw      computr2,computr2,computri      ; 11
1354:      dw      computr3,computr2,computri1      ; 12
1355:      dw      computri1,computr2,computr2      ; 13
1356:      dw      computr2,computr2,computr2      ; 14
1357:      dw      computr3,computr2,computr2      ; 15
1358:      dw      computri1,computr2,computr3      ; 16
1359:      dw      computri1,computr3,computri1      ; 17
1360:      dw      computri1,computr3,computr2      ; 18
1361:      dw      computri1,computr3,computr3      ; 19
1362:      dw      computr3,computr2,computr3      ; 20
1363:      dw      computr3,computr3,computri1      ; 21
1364:      dw      computr3,computr3,computr2      ; 22
1365:      dw      computr3,computr3,computr3      ; 23
1366:      dw      computr2,computr2,computr3      ; 24
1367:      dw      computr2,computr3,computri1      ; 25
1368:      dw      computr2,computr3,computr2      ; 26
1369:      dw      computr2,computr3,computr3      ; 27
1370: ;
1371: init:   lxi      h,2814h
1372:         shld     playco
1373:         lxi      h,2806h
1374:         shld     comco
1375:         shld     memmap1
1376:         lxi      h,2106h
1377:         shld     comco2
1378:         shld     memmap2
1379:         lxi      h,2f06h
1380:         shld     comco3

```

```
1381:     shld    memmap3
1382:     mvi     a, 'u'
1383:     sta     pldir
1384:     mvi     a, '8'
1385:     sta     plmove
1386:     mvi     a, 's'
1387:     sta     combit
1388:     sta     combit2
1389:     sta     combit3
1390:     lxi     h, memmap1+2
1391:     shld   screen1
1392:     lxi     h, memmap2+2
1393:     shld   screen2
1394:     lxi     h, memmap3+2
1395:     shld   screen3
1396:     ret
1397: misc:  lxi     h, 1402h
1398:     call   adj
1399:     lxi     h, score1
1400:     call   show
1401:     lxi     h, 4302h
1402:     call   adj
1403:     call   levout
1404:     lxi     h, 4301h
1405:     call   adj
1406:     call   levouti
1407:     lxi     h, 2001h
1408:     call   adj
1409:     lxi     h, hscore1
1410:     call   show
1411:     lxi     h, 2801h
1412:     call   adj
1413:     lxi     h, name1
1414:     call   show
1415:     lxi     h, 1f02h
1416:     call   adj
1417:     call   cycles
1418:     ret
1419: ;
1420: thought:db  0      ; thought pattern storage
1421: ;
1422: cycles: lda  cycleft ; don't display
1423:         dcr   a       ; 1st cycle in use
1424:         rz
1425: cyclesi:push psw
1426:         mvi   a, 'v'
1427:         call  grout
1428:         pop   psw
1429:         dcr   a
1430:         jnz   cyclesi
1431:         ret
1432: ;
1433: levout: lda  lev1
1434:         adi   30h
1435:         call  output
1436:         ret
1437: ;
1438: levouti:lda  lev2
1439:         inr   a
1440:         cpi   0ah
```

```
1441:     jc      levout2
1442:     lda     lev3
1443:     inr     a
1444:     adi     30h
1445:     call    output
1446:     xra     a
1447:     levout2:adi 30h
1448:     call    output
1449:     ret
1450: ;
1451: dislevel:
1452:     lda     ; see if time to display
1453:     mov     b,a
1454:     lda     lev2
1455:     add     b
1456:     cpi     1      ; test for level 0, round 1
1457:     jz      dislevel1
1458:     lda     levi
1459:     ora     a
1460:     rnz
1461: dislevel1:
1462:     call    graphix; enter graphics mode
1463:     lxi     h,levmes; level message
1464:     call    show
1465: dislevel2:
1466:     lxi     h,370ah
1467:     call    adj     ; display level number
1468:     lda     lev3
1469:     cpi     0      ; 1st digit 0?
1470:     jz      dislevel3
1471:     call    disslecti
1472:     lxi     h,3e0ah
1473:     call    adj     ; next digit position
1474: dislevel3:
1475:     lda     lev2
1476:     inr     a
1477:     call    disslecti
1478:     call    xgraphix; exit graphics mode
1479:     lxi     h,1500h ; delay
1480: disslect:
1481:     dcx
1482:     push   h
1483:     call    input   ; test for input
1484:     pop     h
1485:     ora     a       ; test
1486:     rnz
1487:     mov     a,h
1488:     ora     1
1489:     jnz     disslect
1490:     ret
1491: ;
1492: disslecti:
1493:     cpi     0      ; test for zero
1494:     jz      diszero
1495:     cpi     1
1496:     jz      disone
1497:     cpi     2
1498:     jz      distwo
1499:     cpi     3
1500:     jz      disthree
```

```
1501:      cpi    4
1502:      jz     disfour
1503:      cpi    5
1504:      jz     disfive
1505:      cpi    6
1506:      jz     dissix
1507:      cpi    7
1508:      jz     disseven
1509:      cpi    8
1510:      jz     diseight
1511:      lxi    h,nine
1512:      call   show
1513:      ret
1514:      diseight:
1515:      lxi    h,eight
1516:      call   show
1517:      ret
1518:      disseven:
1519:      lxi    h,seven
1520:      call   show
1521:      ret
1522:      dissix: lxi   h,six
1523:      call   show
1524:      ret
1525:      disfive: lxi  h,five
1526:      call   show
1527:      ret
1528:      disfour: lxi h,four
1529:      call   show
1530:      ret
1531:      disthree:
1532:      lxi    h,three
1533:      call   show
1534:      ret
1535:      distwo: lxi h,two
1536:      call   show
1537:      ret
1538:      disone: lxi h,one
1539:      call   show
1540:      ret
1541:      diszero: lxi h,zero
1542:      call   show
1543:      ret
1544:      ;
1545:      incscor: lxi h,scorei
1546:      inx    h
1547:      inx    h
1548:      inx    h      ; get 100's score
1549:      mov    b,m
1550:      lda    lev
1551:      add    b
1552:      cpi    ':'
1553:      jnc    inc2
1554:      mov    m,a
1555:      jmp    scorout
1556:      inc2: sui   10
1557:      mov    m,a
1558:      ddx    h
1559:      mov    a,m
1560:      inr    a
```

```
1561:      cpi      " ;"
1562:      jnz      xscore
1563:      jmp      inc2
1564:  xscore: lxi      m, a
1565:  scorout: lxi      h, 1402h
1566:          call     adj
1567:          lxi      h, scorei
1568:          call     show
1569:          lhld    freepnt
1570:          lxi      d, scorei
1571:          inx      d      ; 10,000's digit
1572:          ldax    d
1573:          cmp      m
1574:          rnz
1575:          inx      h
1576:          inx      d
1577:          ldax    d
1578:          cmp      m
1579:          rc
1580:          inx      h
1581:          shld    freepnt
1582:          mov      a, m
1583:          cpi      0
1584:          jnz      bonus
1585:          lxi      h, freebie
1586:          shld    freepnt
1587:  bonus:   lda      cycleleft
1588:          inr      a
1589:          sta      cycleleft
1590:  sound:   lxi      h, sndmem
1591:  sound1:  mov      b, m      ; 1st sound
1592:          inx      h
1593:          mov      d, m
1594:          inx      h
1595:          mov      e, m
1596:          inx      h
1597:          mov      c, m
1598:          inx      h
1599:          push    h
1600:          call    sound2
1601:          pop     h
1602:          ret
1603:  sound2:  mvi      a, 10h
1604:          out     0f0h
1605:  sound3:  dcr      b
1606:          jnz      sound3
1607:          mvi      a, 0f4h
1608:          out     0f0h
1609:          push    d
1610:  sound4:  dcx      d
1611:          mov      a, e
1612:          ora     d
1613:          jnz      sound4
1614:          pop     d
1615:          dcr      c
1616:          jnz      sound2
1617:          ret
1618:  pause:   call    wait
1619:          jmp     gamei
1620:  newhigh: lxi      h, 3100h
```

```
1621:          shld    hscore6
1622:          lxi     h,hscore1
1623:  nwhigh: lxi     d,pscore1
1624:          ldax    d
1625:          cmp     m       ; 100,000 position
1626:          jc      nxthigh
1627:          jnz    nhhigh1
1628:  nhhigh2: inx    h
1629:          inx    d
1630:          ldax    d
1631:          cmp     m       ; all other digits
1632:          jc      nxthigh
1633:          jnz    nhhigh1 ;
1634:          jmp    nhhigh2
1635:  nhhigh1: lxi    h,nhmes ; new high message
1636:          call   show
1637:          lxi    h,ieiih
1638:          call   adj    ; display position
1639:          lda    hscore6+1
1640:          call   output
1641:          call   pushdown; push lower scores down.
1642:          call   curhigh
1643:          lxi    d,pscore1
1644:  nhhigh3: ldax   d
1645:          mov    m,a
1646:          inx    h
1647:          inx    d
1648:          mov    a,m
1649:          ora    a
1650:          jnz    nhhigh3 ; repeat until all digits copied
1651:          lxi    h,nhmes1
1652:          call   show
1653:          call   nhnames
1654:          call   zeroname
1655:          call   cursor
1656:          call   cursb
1657:          mvi   m,32
1658:          inx    h
1659:          call   inbuf
1660:          inx    h
1661:          mvi   m,0
1662:          call   cursoff
1663:          ret
1664:  zeroname:
1665:          push   h
1666:          mvi   b,0eh   ; 14 char max
1667:  zeronam1:
1668:          mvi   m,32
1669:          inx    h
1670:          dcr    b
1671:          jnz    zeronam1
1672:          pop    h
1673:          ret
1674:  nxthigh:call  nxthigh1
1675:          jmp    nwhigh
1676:  nxthigh1:
1677:          lhld   hscore6
1678:          inr    h
1679:          shld   hscore6
1680:  curhigh:lhld  hscore6
```

```
1681:     mov    a,h
1682:     sui    31h
1683:     cpi    0
1684:     jz     hscr1
1685:     cpi    1
1686:     jz     hscr2
1687:     cpi    2
1688:     jz     hscr3
1689:     cpi    3
1690:     jz     hscr4
1691:     cpi    4
1692:     jz     hscr5
1693:     cpi    5
1694:             jz     hscr6 ; exit
1695:             ret   ; return to gameov
1696: hscr1: lxi   h,hscore1
1697:         ret
1698: hscr2: lxi   h,hscore2
1699:         ret
1700: hscr3: lxi   h,hscore3
1701:         ret
1702: hscr4: lxi   h,hscore4
1703:         ret
1704: hscr5: lxi   h,hscore5
1705:         ret
1706: hscr6: pop   h      ; get return address
1707:         ret
1708: nhnames:lhld hscore6
1709:         mov   a,h
1710:         sui   31h
1711:         cpi   1
1712:         jz    nhname2
1713:         cpi   2
1714:         jz    nhname3
1715:         cpi   3
1716:         jz    nhname4
1717:         cpi   4
1718:         jz    nhname5
1719: nhname1:lxi h,name1
1720:         ret
1721: nhname2:lxi h,name2
1722:         ret
1723: nhname3:lxi h,name3
1724:         ret
1725: nhname4:lxi h,name4
1726:         ret
1727: nhname5:lxi h,name5
1728:         ret
1729: highscre:
1730:         lxi   h,chart
1731:         call  show
1732:         lxi   h,chart2
1733:         call  show
1734:         lxi   h,1405h
1735:         call  adj
1736:         lxi   h,name1
1737:         call  show
1738:         lxi   h,3205h
1739:         call  adj
1740:         lxi   h,hscore1
```

```
1741:     call    show
1742:     lxi    h, 1407h
1743:     call    adj
1744:     lxi    h, name2
1745:     call    show
1746:     lxi    h, 3207h
1747:     call    adj
1748:     lxi    h, hscore2
1749:     call    show
1750:     lxi    h, 1409h
1751:     call    adj
1752:     lxi    h, name3
1753:     call    show
1754:     lxi    h, 3209h
1755:     call    adj
1756:     lxi    h, hscore3
1757:     call    show
1758:     lxi    h, 140bh
1759:     call    adj
1760:     lxi    h, name4
1761:     call    show
1762:     lxi    h, 320bh
1763:     call    adj
1764:     lxi    h, hscore4
1765:     call    show
1766:     lxi    h, 140dh
1767:     call    adj
1768:     lxi    h, name5
1769:     call    show
1770:     lxi    h, 320dh
1771:     call    adj
1772:     lxi    h, hscore5
1773:     call    show
1774: highscore1:
1775:     call    reverse
1776:     lxi    h, chart1
1777:     call    show
1778:     call    xreverse
1779:     call    input
1780:     ora    a
1781:     jnz    begin
1782:     lxi    h, chart1
1783:     call    show
1784:     call    input
1785:     ora    a
1786:     jnz    begin
1787:     jmp    highscore1
1788: pushdowns:
1789:     lhld   hscode6 ; get counter
1790:     mov    a, h      ;
1791:     sui    30h      ; offset
1792:     cpi    1         ; top score?
1793:     jz     pushall  ; push all scores
1794:     cpi    2         ; second highest
1795:     jz     push2d   ;
1796:     cpi    3         ; third highest
1797:     jz     push3d   ;
1798:     cpi    4         ; fourth place
1799:     jz     push4d   ;
1800:     ret    ; replace 5th
```

```
1801: push1: call nhname1 ; get 1st name
1802:           lxi d, name2 ; destination
1803:           call copyname; copy it
1804:           lxi h, hscore1
1805:           lxi d, hscore2
1806:           call copyscor;
1807:           ret
1808: push2:  call nhname2 ; name to copy
1809:           lxi d, name3 ; where to copy
1810:           call copyname;
1811:           lxi h, hscore2
1812:           lxi d, hscore3
1813:           call copyscor;
1814:           ret
1815: push3:  call nhname3 ;
1816:           lxi d, name4 ;
1817:           call copyname;
1818:           lxi h, hscore3
1819:           lxi d, hscore4
1820:           call copyscor;
1821:           ret
1822: push4:  call nhname4 ;
1823:           lxi d, name5 ;
1824:           call copyname;
1825:           lxi h, hscore4
1826:           lxi d, hscore5
1827:           call copyscor;
1828:           ret
1829: copyname:
1830:           mov a, m      ; get first char
1831:           stax d      ; store it
1832:           inx d      ;
1833:           inx h      ;
1834:           ora a      ; test after copied
1835:           rz       ; return if zero
1836:           jmp copyname; else loop.
1837: copyscor:
1838: copysci: mov a, m      ;
1839:           ora a      ; test for delimiter
1840:           rz       ; return if zero
1841:           stax d      ; store score
1842:           inx d      ;
1843:           inx h      ;
1844:           jmp copysci ;
1845: pushall: call push4  ;
1846:           call push3  ;
1847:           call push2  ;
1848:           call push1  ;
1849:           ret
1850: push2d:  call push4  ;
1851:           call push3  ;
1852:           call push2  ;
1853:           ret
1854: push3d:  call push4  ;
1855:           call push3  ;
1856:           ret
1857: push4d:  call push4  ;
1858:           ret
1859: help:    lxi h, instructions
1860:           call show
```

```

1861:         call    wait
1862:         cpi    cr
1863:         jz     begin
1864:         jmp    help
1865: copyright:
1866:         lxi    h,1717h ; 24th line, 24th column
1867:         call   adj   ; put cursor there
1868:         lda    cbit
1869:         cpi   1
1870:         jnz   cpyrgti
1871: cpyrgt2:lxi h,copyrgt
1872:         call   show  ; print message
1873:         mvi   a,0
1874:         sta    cbit
1875:         ret
1876: cpyrgti:call reverse ;
1877:         lxi    h,copyrgt
1878:         call   show  ;
1879:         mvi   a,1
1880:         sta    cbit
1881:         jmp   xreverse;
1882: ;
1883: ; game grid
1884: ;
1885: grid:  db    ' [CG]@', 3, 1, ' 1ooooooooooooooooooooo'
1886:          db    ' oooooooooooooooooooooo@m'
1887:          db    ' @', 4, 1, ' )@', 4, 80, ' )@', 5, 1, ' )@', 5, 80, ' )'
1888:          db    ' @', 6, 1, ' )@', 6, 80, ' )@', 7, 1, ' )@', 7, 80, ' )'
1889:          db    ' @', 8, 1, ' )@', 8, 80, ' )@', 9, 1, ' )@', 9, 80, ' )'
1890:          db    ' @', 10, 1, ' )@', 10, 80, ' )@', 11, 1, ' )@', 11, 80, ' )'
1891:          db    ' @', 12, 1, ' )@', 12, 80, ' )@', 13, 1, ' )@', 13, 80, ' )'
1892:          db    ' @', 14, 1, ' )@', 14, 80, ' )@', 15, 1, ' )@', 15, 80, ' )'
1893:          db    ' @', 16, 1, ' )@', 16, 80, ' )@', 17, 1, ' )@', 17, 80, ' )'
1894:          db    ' @', 18, 1, ' )@', 18, 80, ' )@', 19, 1, ' )@', 19, 80, ' )'
1895:          db    ' @', 20, 1, ' )@', 20, 80, ' )@', 21, 1, ' )@', 21, 80, ' )'
1896:          db    ' @', 22, 1, ' )@', 22, 80, ' )'
1897:          db    ' @', 23, 1, ' ozzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz'
1898:          db    ' zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz@g1'
1899:          db    ' @', 2, 10, ' SCORE :@', 1, 60, ' Level:@', 2, 60, ' Round:@'
1900:          db    ' @', 1, 16, ' HIGH SCORE :'
1901:          db    ' @', 24, 24, ' Press : <Q> - QUIT / <P> - PAUSE', 0
1902: ;
1903: box1:  lxi    h,490fh ; opposite corner
1904:          shld   boxcol
1905:          lxi    h,708h ; line 8 column 7
1906: box2:  shld   boxco
1907:          call   box3
1908:          lhld   boxcol
1909:          push   h
1910:          call   box3
1911:          pop    h
1912:          mov    a,h
1913:          cpi   7
1914:          rz
1915:          dcr
1916:          shld   boxcol
1917:          lhld   boxco
1918:          inr
1919:          jmp   box2 ; print next block
1920: box7:  lhld   boxco

```

```

1921: box4:    shld   boxco
1922:          call    box3
1923:          lhld   boxcoi
1924:          push   h
1925:          call    box3
1926:          pop    h
1927:          mov    a, l      ; get y
1928:          cpi    8
1929:          rz
1930:          dcr    l
1931:          shld   boxcoi
1932:          lhld   boxco
1933:          inr    l
1934:          jmp    box4
1935: box3:    call    adj
1936:          call    reverse
1937:          lda    boxchar
1938:          call    grout
1939:          call    xreverse
1940:          ret
1941: intro:   db     ' [CG]@', 2, 32, 'fa asaa faaaaaaaaa'
1942:          db     '@', 3, 32, ' '           eac@', 4, 32, 'eaaaadaaaaad'
1943:          db     '@', 3, 33, '[g]lomp@', 3, 37, 'eo@', 3, 43, 'oftware'
1944:          db     '@', 6, 37, 'presents[G]@', 9, 13, '{ {{ @', 9, 31, 't'
1945:          db     '@', 9, 44, '{ { {{ @', 9, 61, 'lme@', 10, 12
1946:          db     'x@', 10, 25, '{ {{ } | { {{ x| y| y { {{ {{ '
1947:          db     ' { {{ @', 11, 12, ' | x| x w x'
1948:          db     ' { {{ | | | | x | | y| x | '
1949:          db     '@', 12, 12, 'y{{{{x y{{xy{{{x y{{w{{{ '
1950:          db     '| yxy{{xyx| }xyxy{{xy@', 13, 13, '{{{{{{{{x[g]@'
1951:          db     '@', 16, 39, 'by@', 18, 36, 'Les Bird@', 2, 62, 'SKILL LEVEL:'
1952:          db     '@', 16, 9, 'Press [R]f1[r] to play CYCLE@', 17, 15
1953:          db     '[R]f2[r] to see high', ap, 's@', 18, 15
1954:          db     '[R]f3[r] for help@', 16, 49
1955:          db     'Press [R]f4[r] to select speed@', 17, 49
1956:          db     ' [R]f5[r] to select level@', 18, 49
1957:          db     ' [R]blue[r] to quit', 0
1958: chart:    db     ' [C]@', 2, 29, '[R] C Y C L E M A N I A [r]'
1959:          db     '@', 3, 30, 'ALL TIME HIGH SCORES'
1960: chart1:   db     '@', 4, 18, '[G]faaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
1961:          db     'aaaaaaaaaaaaac@', 5, 18, '@', 5, 62, ''
1962:          db     '@', 6, 18, '@', 6, 62, '@', 7, 18, '@', 7, 62
1963:          db     '@', 8, 18, '@', 8, 62, '@', 9, 18, '@', 9, 62
1964:          db     '@', 10, 18, '@', 10, 62, '@', 11, 18, ''
1965:          db     '@', 11, 62, '@', 12, 18, '@', 12, 62, ''
1966:          db     '@', 13, 18, '@', 13, 62, '@', 14, 18
1967:          db     'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaad[g]', 0
1968: chart2:   db     '@', 5, 16, '1.@', 7, 16, '2.@', 9, 16, '3.@', 11, 16, '4.'
1969:          db     '@', 13, 16, '5.', 0
1970: mess10:   db     '[CR]@', 2, 25, ' C Y C L E M A N I A [r]'
1971:          db     '[r]@', 5, 20, 'Select speed level :'
1972:          db     ' [R]f1[r] for expert@', 6, 41, '[R]f2[r] for intermediate'
1973:          db     '@', 7, 41, '[R]f3[r] for level 1 intermediate@', 8, 41
1974:          db     '[R]f4[r] for novice', 0
1975: nhmes:    db     '[CG]@', 8, 18
1976:          db     esc, 'jfaac', esc, 'k', lf, ' ', bs, lf, 'eaad@', 9, 23 ; C
1977:          db     esc, 'jfc', esc, 'k', lf, 'ed@', 9, 26 ; O
1978:          db     esc, 'jsc', esc, 'k', lf, 'ee@', 9, 29 ; N
1979:          db     esc, 'jfc', esc, 'k', lf, 'et@', 11, 18, 'aaaaaaaaaaaa' ; G
1980:          db     '@', 9, 32

```

```

1981: db      esc, 'Jfa', esc, 'k', lf, 'e@', 9, 35      r
1982: db      esc, 'Jfc', esc, 'k', lf, 'eu@', 9, 38      a
1983: db      esc, 'Jft', esc, 'k', lf, 'eu@', 8, 39, '@', 9, 41      d
1984: db      esc, 'Jff', esc, 'k', lf, 'ed@', 8, 44      u
1985: db      esc, 'Jc', esc, 'k', lf, '@', 10, 44, 'e@', 9, 46      l
1986: db      esc, 'Jfc', esc, 'k', lf, 'eu@', 8, 50      a
1987: db      '@', 9, 49, 'aba@', 10, 50, 'e@', 9, 53      t
1988: db      esc, 'js', esc, 'k', lf, 'u@', 9, 55      i
1989: db      esc, 'Jfc', esc, 'k', lf, 'ed@', 9, 58      o
1990: db      esc, 'jsc', esc, 'k', lf, 'ee@', 9, 61      n
1991: db      esc, 'jf', esc, 'k', lf, 'd[g]',      s
1992: db      '@', 14, 21, 'You made the ALL TIME HIGH SCORE CHART'      s
1993: db      '@', 15, 22, 'Enter your name, 14 character limit.'      n
1994: nhmesi: db      '@', 17, 31, '____@', 17, 32, 0      s
1995: ;
1996: expmes: db      '@', 3, 64, '[G]1[R]pppppp[r]m@', 4, 64, '[R]q[rg]EXPERT'
1997: db      '[G]q@', 5, 64, 'oppppppn[g]', 0
1998: promes: db      '@', 3, 66, '[G]{{{{[R]}@', 4, 66, 'q[r]PROq@', 5, 66, 'zzzzz[g]'}

,0
1999: intmes: db      '@', 3, 62, '[G] {{{{{{{{{@', 4, 62, 'INTERMEDIATE'
2000: db      '@', 5, 62, 'zzzzzzzzzz [g]', 0
2001: amames: db      '@', 3, 62, '@', 4, 64, 'AMATUERO', 5, 64, 0
2002: copyrgt:db      '<C>opyright 1983 CompTec Software', 0
2003: instructions:
2004:         db      '[C]@', 1, 25, '[R] C Y C L E M A N I A [r]'
2005:         db      '@', 2, 25, '[G]aaaaaaaaaaaaaaaaaaaaaa[g]'
2006:         db      0
2007: ;
2008: cycle19:call    cls
2009:         call    clr25
2010: cycle0: lxi     h, 4f19h ; column 79, line 25
2011:         call    adj
2012:         mvi     a, 25
2013:         sta     misbit
2014:         lxi     h, cyclemes
2015: cycle1: mov     a, m
2016:         cpi     0
2017:         jz      cycle0
2018:         push   h
2019:         call    output
2020:         lxi     h, 119h ; x=1, y=25
2021:         call    adj
2022:         call    deletec
2023:         lxi     h, 4f19h
2024:         call    adj
2025:         lxi     h, 2500h
2026: cycle2: dcx     h
2027:         mov     a, h
2028:         ora     1
2029:         jnz     cycle2
2030:         call    input
2031:         ora     a
2032:         pop     h
2033:         jnz     cycle3
2034:         inx     h
2035:         jmp     cycle1
2036: cycle3: call    clr25
2037:         jmp     begin
2038: ;
2039: ; open or create high score file
2040: ;

```

```
2041: highfile:
2042:         call      openfile; open the file
2043:         call      readfile; read data
2044:         ora      a
2045:         rnz
2046: ; read high scores from disk
2047:         lxi      b,0580h ; five high scores
2048:         lxi      h,dma
2049: highi: lxi      d,name1 ; copy name *** storage location
2050: highfilei:
2051:         mov      a,m
2052:         dcr      c
2053:         cz      highfile6
2054:         cpi      '='      ; name/score separator
2055:         jz      highfile2
2056:         stax      d
2057:         imx      h
2058:         inx      d
2059:         jmp      highfile1
2060: highfile2: ; ***** storage location *****
2061:         lxi      d,hscore1
2062:         inx      h
2063: highfile3:
2064:         mov      a,m      ; get score
2065:         dcr      c
2066:         cz      highfile6
2067:         cpi      '/'      ; load next name/score
2068:         jz      highfile4
2069:         cpi      '*'      ; end of file
2070:         rz
2071:         stax      d      ; store in score
2072:         inx      h
2073:         inx      d
2074:         jmp      highfile3
2075: highfile4:
2076:         inx      h      ; start of next name
2077:         dcr      b      ; count=count-1
2078:         rz
2079:         mov      a,b
2080:         cpi      1
2081:         jz      high5
2082:         cpi      2
2083:         jz      high4
2084:         cpi      3
2085:         jz      high3
2086: high2: push      h
2087:         lxi      h,name2
2088:         shld     high1+i
2089:         lxi      h,hscore2
2090:         shld     highfile2+i
2091:         pop      h
2092:         jmp      highi
2093: high3: push      h
2094:         lxi      h,name3
2095:         shld     high1+i
2096:         lxi      h,hscore3
2097:         shld     highfile2+i
2098:         pop      h
2099:         jmp      high1
2100: high4: push      h
```

```
2101:      lxi    h,name4
2102:      shld   highi+1
2103:      lxi    h,hscore4
2104:      shld   highfile2+1
2105:      pop    h
2106:      jmp    highi
107: high5: push   h
2108:      lxi    h,name5
2109:      shld   highi+1
2110:      lxi    h,hscore5
2111:      shld   highfile2+1
2112:      pop    h
2113:      jmp    highi
2114: highfile6:
2115:      call   savall
2116:      call   readfile
2117:      call   retall
2118:      mvi   c,80h
2119:      lxi    h,dma
2120:      ret
2121: ; write high scores to disk
2122: wrhigh: call   delfile
2123:      call   makefile
2124:      lxi    d,dma
2125:      lxi    b,0580h ; five high scores
2126: whigh6: lxi    h,name1 ; *** storage location ***
2127: wrhigh1: mov   a,m
2128:      stax   d
2129:      dcr    c
2130:      cz    wrhigh6
2131:      ora    a
2132:      jz    wrhigh2
2133:      inx   h
2134:      inx   d
2135:      jmp    wrhigh1
2136: wrhigh2:inx  d
2137:      mvi   a,'=' ; name/score separator
2138:      stax   d ; put in DMA
2139:      dcr    c
2140:      cz    wrhigh6
2141:      inx   d
2142: whigh7: lxi    h,hscore1
2143: wrhigh3: mov   a,m
2144:      stax   d
2145:      dcr    c
2146:      cz    wrhigh6
2147:      ora    a
2148:      jz    wrhigh4
2149:      inx   h
2150:      inx   d
2151:      jmp    wrhigh3
2152: wrhigh4:mvi a,'/' ; *** storage location ***
2153:      inx   d
2154:      stax   d
2155:      dcr    b
2156:      jz    wrhigh7
2157:      dcr    c
2158:      cz    wrhigh6
2159:      inx   d
2160:      mov   a,b
```

```

2161:      cpi    1
2162:      jz     whigh5
2163:      cpi    2
2164:      jz     whigh4
2165:      cpi    3
2166:      jz     whigh3
167: whigh2: lxi   h,name2
2168:      shld   whigh6+1
2169:      lxi   h,hscore2
2170:      shld   whigh7+1
2171:      jmp    whigh6
2172: whigh3: lxi   h,name3
2173:      shld   whigh6+1
2174:      lxi   h,hscore3
2175:      shld   whigh7+1
2176:      jmp    whigh6
2177: whigh4: lxi   h,name4
2178:      shld   whigh6+1
2179:      lxi   h,hscore4
2180:      shld   whigh7+1
2181:      jmp    whigh6
2182: whigh5: lxi   h,name5
2183:      shld   whigh6+1
2184:      lxi   h,hscore5
2185:      shld   whigh7+1
2186:      mvi    a,'*' ; end of file
2187:      sta    wrhigh4+1
2188:      jmp    whigh6
2189: wrhigh6:call savall ; all registers on stack
2190:      call   writefile
2191:      call   retall
2192:      mvi    c,80h ; another 128 bytes
2193:      lxi   d,dma ; reset DMA
2194:      ret
2195: wrhigh7:dcr c
2196:      jz     wrhigh6 ; write file
2197:      mvi    a,20h ; space
2198:      inx
2199:      stax
2200:      jmp    wrhigh7
2201: cyclemes:
2202:      db     ' (C)opyright 1983 Comp'
2203:      db     ' Tec Software by'
2204:      db     ' Les Bird. . . . Press any key for main menu'
2205:      db     ' CYCLE MANIA ',0
2206: cyclemes1:
2207:      db     'CYCLE MANIA',0
2208: levmes: db     '@',10,19
2209:      db     '[R]r[r]r [R]r[r]ppp[R]_[r] [R]r[r] [R]_[r]'
2210:      db     '[R]r[r]ppp[R]_[r] [R]r[r]r'
2211:      db     '@',11,19
2212:      db     '[R]_[r] [R]_[r] _[R]_[r] [R]r[r]r '
2213:      db     '[R]_[r] [R]_[r] '
2214:      db     '@',12,19
2215:      db     '[R]_[r] [R]_[r]ppp _[R]_r[r]r '
2216:      db     '[R]_[r]ppp [R]_[r] '
2217:      db     '@',13,19
218:      db     '_[R]ppp[r]r _[R]ppp[r]r _r _[R]ppp[r]r _[R]ppp[r]r'
,0
2219: levmes1:db     '@',8,28,'[g]Enter level: 1=easy, 9=hard[G]',0
2220: one:   db     '1bh,'j [R]r [rc]',1bh,'k',1f

```

```

2221:      db      ibh,'j [r] [rc]',ibh,'k',lf
2222:      db      ibh,'j [r] [rc]',ibh,'k',lf
2223:      db      ' [r]p p[r] ',0
2224: two:   db      ibh,'j [r]r[r]pppp[R]_[rc]',ibh,'k',lf
2225:      db      ibh,'j [r]pppp[r]r[c]',ibh,'k',lf
2226:      db      ibh,'j [r]r[rc]',ibh,'k',lf
2227:      db      ' [r] pppp[rc]',0
2228: three: db      ibh,'j [r]r[r]pppp[R]_[rc]',ibh,'k',lf
2229:      db      ibh,'j {{[r] [rc]',ibh,'k',lf
2230:      db      ibh,'j zzz[R] [rc]',ibh,'k',lf
2231:      db      '_[r]pppp[r]r[c]',0
2232: four:  db      ibh,'j [r]r[r] [r]_[rc]',ibh,'k',lf
2233:      db      ibh,'j [r] [r] [rc]',ibh,'k',lf
2234:      db      ibh,'j pppp[R] [r]p[c]',ibh,'k',lf
2235:      db      ' [r] [rc]',0
2236: five:  db      ibh,'j [r] [r]ppppp[c]',ibh,'k',lf
2237:      db      ibh,'j [r] ppppp[rc]',ibh,'k',lf
2238:      db      ibh,'j [r] [rc]',ibh,'k',lf
2239:      db      '_[r]ppppp[r]r[c]',0
2240: six:   db      ibh,'j [r]r[r]pppp[R]_[rc]',ibh,'k',lf
2241:      db      ibh,'j [r] [r] [c]',ibh,'k',lf
2242:      db      ibh,'j [r] [r]pppp[R]_[rc]',ibh,'k',lf
2243:      db      '_[r]ppppp[r]r[c]',0
2244: seven: db      ibh,'j [r]r[r]ppp[R] [r]r[c]',ibh,'k',lf
2245:      db      ibh,'j [r]r[r]r[c]',ibh,'k',lf
2246:      db      ibh,'j [r]r[r]r[c]',ibh,'k',lf
2247:      db      ' [r]r[r]r[c]',0
2248: eight: db      ibh,'j [r]r[r]pppp[R]_[rc]',ibh,'k',lf
2249:      db      ibh,'j {{[r][c]',ibh,'k',lf
2250:      db      ibh,'j [r]r[r]zzz[R]_[rc]',ibh,'k',lf
2251:      db      '_[r]pppp[r]r[c]',0
2252: nine:  db      ibh,'j [r]r[r]pppp[R]_[rc]',ibh,'k',lf
2253:      db      ibh,'j _[r]pppp [rc]',ibh,'k',lf
2254:      db      ibh,'j [r] [rc]',ibh,'k',lf
2255:      db      '_[r]ppppp[r]r[c]',0
2256: zero:  db      ibh,'j [r]r [r]ppp[R]_[rc]',ibh,'k',lf
2257:      db      ibh,'j [r] [r]_[r]_[r] [rc]',ibh,'k',lf
2258:      db      ibh,'j [r] [r] _[r]_[rc]',ibh,'k',lf
2259:      db      '_[r]ppp [r]r[c]',0
2260: ;
2261: ; error messages
2262: ;
2263: nserror:db      '@',12,5,'[r]I think you should allocate more disk space
',0
2264: nferror:db      '@',12,5,'[r]It seems I am trying to open a file that'
2265:      db      ' does not exist.',0
2266: rerror: db       '@',12,5,'[r]In order for me to read from this disk, you
2267:      db      ' must BOOT UP on it.',0
2268: wrerror:db      '@',12,5,'[r]I think you need to BOOT UP on this disk so
2269:      db      ' that I can write on it.',0
2270: ;
2271: ; disk equates
2272: ;
2273: drive   db      0      ; select current drive
2274: extnum  db      0
2275: reccont db      0
2276: currec  db      0
2277: recnum  db      0
2278: filename: db      'CYCLE19 ',0
2279: filetype: db

```

```
2281:           db      'DAT', 0
2282: ;
2283: retvec: db      0, 0
2284: gbit: db      0
2285: rbit: db      0
2286: cbit: db      0
287: misbit: db      0
2288: crtbit: db      0
2289: graphx: db      0
2290: kpad: db      0
2291: ;          B      D      E      C      ; Free cycle soundfx
2292: sndmem: db      033h, 000h, 05dh, 0ffh ; 1st sound
2293: ;
2294: oldstack: ds      64h
2295: stack: ds      64h
2296: cycleleft:db      4
2297: char1: db      0
2298: misco: ds      2
2300: curco: db      1, 1
2301: curco1: db      1, 1
2302: boxco: db      1, 1
2303: boxco1: db      1, 1
2304: boxchar:db      'i'
2305: linco: db      1, 1
2306: linco1: db      1, 1
2307: linchar:db      'i'
2308: plac0:
2309: playco:
2310: playco1:db      1, 1
2311: plmdir:
`312: plabit:
2313: plabit1:ds      1
2314: plmove:
2315: plamove:
2316: plmove1:db      '8'
2317: speed:
2318: speed1: db      1
2319: speed2: db      1
2320: speed3: db      1
2321: comco:
2322: comco1: db      1, 1
2323: comco2: db      1, 1
2324: comco3: db      1, 1
2325: comdir:
2326: combit:
2327: combit1:db      1
2328: combit2:db      1
2329: combit3:db      1
2330: combit4:db      1
2331: cdbiti: db      0
2332: cdbit2: db      1
2333: cdbit3: db      1
2334: cspeed1:db      0
2335: oldco:
2336: oldco1: db      1, 1
`337: wall:
`338: wall1: ds      1
2339: pscore1:
2340: score:
```

```
2341: score1: db      '00000000', 0
2342: names:
2343: name1: db      1, ' [R]C[R] o m p [R]T[R] e c ', 0
2344: name2: db      1, ' [R]C[R] o m p [R]T[R] e c ', 0
2345: name3: db      1, ' [R]C[R] o m p [R]T[R] e c ', 0
2346: name4: db      1, ' [R]C[R] o m p [R]T[R] e c ', 0
2347: name5: db      1, ' [R]C[R] o m p [R]T[R] e c ', 0
2348: hscores:
2349: hscore1:db     '015000', 0
2350: hscore2:db     '013000', 0
2351: hscore3:db     '009000', 0
2352: hscore4:db     '005000', 0
2353: hscore5:db     '001200', 0
2354: hscore6:ds     2
2355: freepnt:ds    2
2356: ; FREEBIE is list of free cycle points
2357: ; ex. 15 = 15,000/30 = 30,000 etc.
2358: freebie:db     '153045607590', 0
2359: time: db        00, 06
2360: level: db       1
2361: lev: db         1
2362: lev1: db        1
2363: lev2: db        0
2364: lev3: db        0
2365: skill: db       3
2366: scrpnt: db      1, 1
2367: screen1:ds     2
2368: screen2:ds     2
2369: screen3:ds     2
2370: screen4:ds     2
2371: memmap1:ds     1000
2372: memmap2:ds     1000
2373: memmap3:ds     1000
2374: memmap:
```