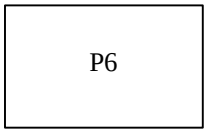
Towers of Hanoi documentation

Program tape:

Space PK

T 56 K



Space PZ

E Z P F

Table of routines:

Routine	Location of first order	Number of storage locations used
P6	56	36
Master	88	-

Master routine:

		GK	
		T47K	
		P156@	
		TZ	
	0	O1M	
	1	O1M	Prepare for output
	2	O2M	
	3	O3M	
	4	T80M	
	5	A5M	
	6	T13M	
	7	A12M	
	8	T14M	
	9	A4M	Store initial values
	10	T15M	MOVE(n, position 1, pile 1, pile 3, pile 2)
	11	A6M	
	12	T16M	
	13	A5M	
	14	T17M	
			Start of MOVE (number, called from, from pile , to pile , other pile)
74, 75, 87, 88, 101, 102→	15	T80M	Clear the accumulator
	16	T78M	Set the counter to zero
35→	17	T80M	Clear the accumulator
	18	A86M	
	19	A87M	Form a transfer to top of stack command
	20	T22@	Place the transfer command in the line 2 below this

```
          21    A13M
          22    PF        Transfer a stored value onto the stack
          23    A19@
          24    A5M       Increment the 'A(n)M' command in position 19@ to 'A(n+1)M'
          25    T19@
          26    A21@
          27    A5M       Increment the 'A(n)M' command in position 21@ to 'A(n+1)M'
          28    T21@
          29    A78M
          30    A5M       Increment the counter by 2
          31    T78M
          32    A78M
          33    S11M [-8] If the counter is less than 8, then repeat this loop
          34    S4M  [-1]
          35    G17@
          36    T80M      Clear the accumulator
          37    A86M
          38    A11M      Increment the stack pointer
          39    A5M
          40    T86M
          41    A97M
          42    T19@      Replace the altered lines of code in positions 19@ and 21@
          43    A98M
          44    T21@
                          Data has now been pushed onto the stack
          45    T80M
          46    A14M      If the number of counters to move is more than 1, then skip the
          47    S6M       following 13 lines
          48    E62@
          49    T80M      Clear the accumulator
          50    A15M
          51    TF        Print the 'from' number
          52    A52@
          53    G56F
          54    A16M
          55    TF        Print the 'to' number
          56    A56@
          57    G56F
          58    O2M       Print end-of-line
          59    O3M
          60    G103@     Skip the rest of this code and go to the return section
          61    E103@
                ----------
48→       62    T80M      Clear the accumulator
          63    A7M
          64    T13M
          65    A14M
          66    S5M
          67    T14M
          68    A17M      Store information in preparation for beginning the MOVE section
          69    T85M      MOVE(n-1,position 2, from pile, other pile, to pile)
          70    A16M
          71    T17M
          72    A85M
          73    T16M
          74    G15@      Go to the beginning of the MOVE section
          75    E15@
                ----------

148→      76    T80M      Clear the accumulator
```

	77	A9M	Store information in preparation for beginning the MOVE section MOVE(1,position 3, from pile, to pile, other pile)
	78	T13M	
	79	A5M	
	80	T14M	
	81	A15M	
	82	T15M	
	83	A16M	
	84	T16M	
	85	A17M	
	86	T17M	
	87	G15@	Go to the beginning of the MOVE section
	88	E15@	
152→	89	T80M	Clear the accumulator
	90	A11M	Store information in preparation for beginning the MOVE section MOVE(n-1,position 4, other pile, to pile, from pile)
	91	T13M	
	92	A14M	
	93	S5M	
	94	T14M	
	95	A17M	
	96	T85M	
	97	A15M	
	98	T17M	
	99	A85M	
	100	T15M	
	101	G15@	Go to the beginning of the MOVE section
	102	E15@	
			Return section
60, 61, 153,154→	103	T80M	Clear the accumulator
	104	A86M	Decrement the stack pointer
	105	S11M	
	106	S5M	
	107	T86M	
	108	A92M	Create an ‘add number from top of stack’ command
	109	A86M	
	110	T111@	Keep the ‘position’ variable safe because it is needed later
	111	PF	
	112	T85M	
	113	T80M	Clear the accumulator
	114	T78M	Zero the counter
135→	115	T80M	Clear the accumulator
	116	A86M	Take the stack pointer - one record
	117	S11M	
	118	S5M	
	119	A92M	Combine it to form an ‘add number from stack’ command
	120	T121@	Place this command in the line below
	121	PF	Store the number from the stack
	122	T13M	
	123	A119@	Increment the command in 119@ so that it points to a different part of the stack
	124	A5M	
	125	T119@	
	126	A122@	Increment the command in 122@ so that it points to a different storage location
	127	A5M	
	128	T122@	
	129	A78M	Increment the counter
	130	A5M	
	131	T78M	

```
                132     A78M
                133     S11M [-8]   If the counter  is less than 8 then repeat this section
                134     S4M  [-1]
                135     G115@
                136     T80M        Clear the accumulator
                137     A99M
                138     T119@       Replace the altered lines of code in 119@ and 122@
                139     A100M
                140     T122@
                141     T80M        Clear the accumulator
                142     A85M        Check if the position MOVE was called from was position 1.
                143     S6M         If so, then go to the end of the program
                144     G155@
                145     T80M        Clear the accumulator
                146     A85M
                147     S8M         If MOVE was called from position 2, return to the appropriate place
                148     G76@
                149     T80M        Clear the accumulator
                150     A85M
                151     S10M        If MOVE was called from position 3, return to the appropriate place
                152     G89@
                153     G103@       MOVE must have been called from position 4, so return to the
                154     E103@       appropriate place
144→            155     ZF          STOP
          M     0       PF          Padding
                1       #F
                2       @F          Characters for printing
                3       &F
                4       P0D         =1
                5       P1F         =2
                6       P1D         =3
                7       P2F         =4
                8       P2D         =5
                9       P3F         =6
                10      P3D         =7
                11      P4F         =8
                12      P7F         NUMBER OF COUNTERS TO USE * 2 (7 counters used here)
                                    DO NOT SET TO GREATER THAN P12F
                13      PF
                14      PF
                15      PF          Temporary storage locations
                16      PF
                17      PF
                18      PF
                19      PF
                20      PF
                21      PF
                22      PF
                23      PF
                24      PF
                25      PF
                26      PF          Stack
                27      PF
                28      PF
                29      PF
                30      PF
                31      PF
                32      PF
                33      PF
                34      PF
```

35	PF	
36	PF	
37	PF	
38	PF	
39	PF	
40	PF	
41	PF	
42	PF	
43	PF	
44	PF	
45	PF	
46	PF	
47	PF	
48	PF	
49	PF	
50	PF	
51	PF	
52	PF	
53	PF	
54	PF	
55	PF	
56	PF	More stack
57	PF	
58	PF	
59	PF	
60	PF	
61	PF	
62	PF	
63	PF	
64	PF	
65	PF	
66	PF	
67	PF	
68	PF	
69	PF	
70	PF	
71	PF	
72	PF	
73	PF	
74	PF	
75	PF	
76	PF	
77	PF	
78	PF	=0
79	P55D	(SIZE OF STACK - 5) * 2 + 1= 111
80	PF	Used to clear the accumulator to
81	A72M	
82	T77M	Stores commands needed to replace altered code
83	A23M	
84	T18M	
85	PF	Used as a counter
86	PF	Stack pointer

87	T18M	Used with the stack pointer to construct commands related to the stack
88	T19M	
89	T20M	
90	T21M	
91	T22M	
92	A18M	
93	A19M	
94	A20M	
95	A21M	
96	A22M	
97	A87M	Used to replace altered code
98	A13M	
99	A92M	
100	T13M	

Description

Outputs the Towers of Hanoi solution in the form:

```
1       3
1       2
3       2
1       3
2       1
2       3
1       3
1       2
3       2
3       1
2       1
…
```

Where the first integer on the row is the pile to take a counter from, and the second integer is the pile to place it one. The solution always moves the entire pile from 1 to 3.

Will work for any number of counters from 1 to 12. Set the value at location 12M in order to change the number of counters used. P*n*F uses *n* counters. For example to have 9 counters instead of 7, change the value from P7F to P9F.

The program is structured as a recursion. The bulk of the program is the function which moves a pile of counters from one place to another, and a stack is used to store function information and local variables. This is not the simplest way to implement the Towers of Hanoi solution (an iterative solution would be easier to program on the EDSAC), but it does demonstrate how a high-level programming concept (recursion) can be implemented at this low level, simply by programming a stack.

Note on numbers in this program. A lot of numbers are represented as twice their actual value. This facilities reading (e.g. P5F represents 5) and also checking the values of the numbers, and obviously does not affect the functionality of the code.

Daniel Tebbutt
January 2001