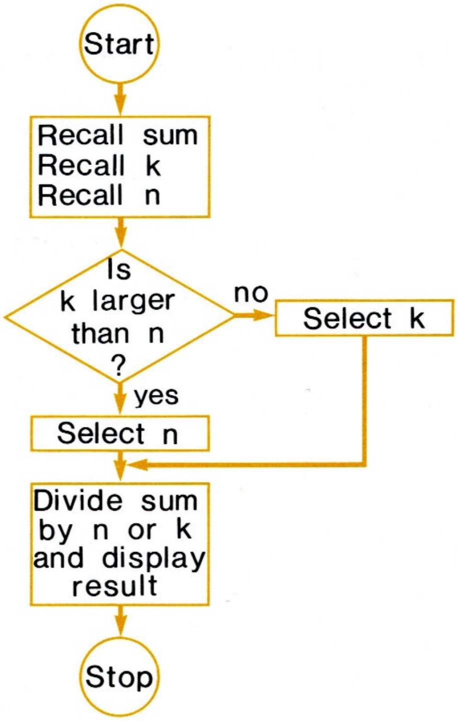
COMPARISON

Subroutine D of *Moving Average* computes the moving average when the D key is pressed from the keyboard.

```
LBL D
RCL 0
RCL E
RCL D
x≤y?
x⇄y
R↓
÷
RTN
```

Generally, the average is calculated based on the summation of input values, Σ (stored in R_0) and the requested number of units, n (stored in R_D) in the moving average. However, if less than n values have been input, the average must be calculated based on the current number of inputs (k). The value of k is stored in R_E. The flowchart for this calculation might look like this:



Subroutine D begins by recalling the sum from R_0, k from R_E and n from R_D. After these recalls the operational stack is as follows:

Unknown value	T
Sum	Z
k	Y
n	X

The comparison step $x \leqslant y$ (if x is less than or equal to y) causes program execution to *skip* the next step when the conditions of the comparison are *not met*. If the conditions of the comparison are met, the*following step is executed.* This is the ''*DO* if *TRUE*'' rule. For instance, if k = y = 15 and n = x = 6 the comparison would be true or satisified (since x is less than y) and the next step, x⇄y (x exchange y), would be executed. If k were less than 6, say 4, the x⇄y command would be skipped. The stack contents for both cases are shown below:

BEFORE COMPARISON

Unknown value	T	Unknown value	T
Sum	Z	Sum	Z
15	Y	4	Y
6	X	6	X

AFTER COMPARISON AND NEXT STEP

Unknown value		T	Unknown value		T
Sum		Z	Sum		Z
6	switched	Y	4	not switched	Y
15		X	6		X

The next step rolls the stack down removing the unwanted value from the X-register.

15 (Unwanted value)	T	6 (Unwanted value)	T
Unknown value	Z	Unknown value	Z
Sum	Y	Sum	Y
6	X	4	X

The last step divides the sum by the value in the X-register to complete the calculation.

L01-03

Moving Average

001	*LBLa		057	R↓	
002	CLRG	Clear registers.	058	RTN	
003	P⇄S		059	*LBL6	If print mode is off pause for display of n.
004	CLRG		060	X⇄Y	
005	1	If $1 \leqslant n \leqslant 22$ continue, otherwise go to label 1.	061	F0?	
006	X>Y?		062	GTO0	
007	GTO1		063	PSE	
008	CLX		064	*LBL0	
009	- 2		065	RCL0	Compute average.
010	2		066	RCLD	
011	X⇄Y		067	÷	
012	X>Y?		068	ENT↑	Output and set for display.
013	GTO1		069	F0?	
014	STOD		070	PRTX	
015	1	Store n in R_D and (n + n/100) in R_I.	071	RTN	Write data.
016	%		072	*LBLE	
017	+		073	WDTA	
018	STOI		074	RTN	
019	INT		075	*LBLb	
020	RTN		076	F0?	Print/pause mode toggle.
021	*LBL1		077	GTO6	
022	R↓	Flash input error.	078	1	
023	*LBL4		079	SF0	
024	PSE		080	RTN	
025	GTO4		081	*LBL6	
026	*LBLA	Increment k by one. Print space, k, and input if flag 0 is set.	082	0	
027	F0?		083	CF0	
028	SPC		064	RTN	
029	RCLE		085	*LBLC	
030	1		086	SPC	Output values in newest to oldest order.
031	+		087	0	
032	F0?		088	*LBL3	
033	PRTX		089	RCLD	
034	X⇄Y		090	X=Y?	
035	F0?		091	RTN	
036	PRTX		092	1	
037	RCLi	Remove oldest value from sum and add input.	093	%	
038	ST-0		094	+	
039	X⇄Y		095	RCLI	
040	STOi		096	X=Y?	
041	ST+0		097	FRC	
042	R↓	Store k.	098	STOI	
043	X⇄Y		099	ISZI	
044	STOE		100	RCLi	
045	RCLD	If n ⩽ k,GTO 0 and calculate average.	101	PRTX	
046	X≤Y?		102	R↑	
047	GSB0		103	1	
048	DSZI	If I is not zero, GTO 5 for display	104	+	
049	GTO5		105	GTO3	
050	RCLI		106	*LBLD	
051	1		107	RCL0	Compute average at any time.
052	0	Reset index for another loop.	108	RCLE	
053	1		109	RCLD	
054	×		110	X≤Y?	
055	STOI		111	X⇄Y	
056	*LBL5	Display average or n.	112	R↓	

REGISTERS

0 Σ	1 used	2 used	3 used	4 used	5 used	6 used	7 used	8 used	9 used
S0 used	S1 used	S2 used	S3 used	S4 used	S5 used	S6 used	S7 used	S8 used	S9 used
A used	B used	C used	D n	E k	I control				

```
   113     ÷          -24
   114    RTN          24
   115    R/S          51
```

LABELS					FLAGS	SET STATUS		
A x→"k," Avg	B W DATA	C →VAL	D →AVG	E	0 print	FLAGS	TRIG	DISP
a n	b P?	c	d	e	1	ON OFF 0 □ ☒ 1 □ ☒ 2 □ ☒ 3 □ ☒	DEG ☒ GRAD □ RAD □	FIX ☒ SCI □ ENG □ n 2
0 used	1 error	2	3 print	4 error	2			
5 display	6	7	8	9	3			