

---

 Programs

Loop address RAM	>100 FOR R=_32768 to 32767
Load that 8 bytes into space	>110 CALL MOVES("RV",8,R,1024)
Loop back	>120 NEXT R
Line 100 sets string-variable.	>100 I\$=RPT\$("I",255)
Line 120 type\$ specifies I\$ to VDP. 55 bytes are moved.	>110 CALL MOVES("\$V",55,I\$,0)
Line 120 copies string J\$ to into lower 8K, then string I\$ into lower 8K.	>120 CALL MOVES("\$R",255,J\$,8192,"\$R",255,I\$,8492)
Line 130 copies string I\$ to into J\$. Eliminates old J\$. Then prints them.	>130 J\$=I\$ :: PRINT J\$ : : I\$
Line 150 copies from lower 8K to J\$, then from lower 8K at 8492 into I\$ thus restoring both strings and printing them thus a way to save stings.	>140 CALL MOVES("R\$",255,8192,J\$,"R\$",255,8492,I\$) :: PRINT J\$: :I\$
Line 100 sets up loop. Counts from -32768 to 0 to 32767 or (HEX >8000 to >0000 to >7FFF)	>100 FOR GA=-32768 TO 32767
Line 110 moves type\$ GRAM/GROM to VDP. 8 bytes to be moved. GA is counter. H\$ is string for storing data found.	>110 CALL MOVES("G\$",8,GA,H\$)
Line 120 prints H\$ on screen.	>120 PRINT H\$
Line 130 next loop	>130 NEXT GA

## Options

Dependent on Assembly Language programmers and the RXB programs that are developed. MOVES is good for replacing those CALL LOAD loops. It also provides a means to rewrite XB while running XB instead of rewriting MERGE files then loading them. Future devices benefit from MOVES as it can copy or move different types of memory directly from or to them.