

```

><><   TRICKS OF THE TRADE   ><><
><><><><><   TUTORIAL #13   ><><><><><
><><><><><><   MIKE WHITE   ><><><><><><
><><><><><><   COUNTY LINE #2   ><><><><><><
><><><><><><   R.D.#1 BOX 373   ><><><><><><
><><><><><><   WAKEMAN , OHIO   ><><><><><><
><><><><><><   44889   ><><><><><><

```

STRINGS

BEFORE WE LOOK AT MULTIPLE ARRAYS, I WISH TO DISCUSS THE TWO TYPES OF STRINGS MENTIONED LAST MONTH.

"FIXED" ARRAYS ARE THOSE WHERE THE AUTHOR FILLS A STRING UP WITH NUMBERS AND TAPES THE ARRAY WITH THE PROGRAM. THE PROGRAM READS DATA OUT OF THE ARRAY, BUT NEVER ERASES OR WRITES INTO THE ARRAY. IF THE ARRAY GETS COPIED TO ANOTHER LOCATION THE "FIXED" PORTION REMAINS. A TOTALLY "MOVING" ARRAY (WHERE THE "FIXED" PORTION DISSAPPEARS AFTER THE MOVE) IS VERY RARE. MY COLLECTION OF BALLY SOFTWARE IS FAR FROM COMPLETE, BUT I CAN FIND NO SAMPLE OF A "MOVING" ARRAY, ANYWHERE!

THE OTHER TYPE IS THE "AREA". SCORES ETC. FORM THESE STRINGS WHICH GET ERASED, WRITTEN INTO, READ, AND BECOME "MOVING" ON OCCASION. SEE QUADRA AB, AND REREAD "TIME SHARING" (LAST MONTHS "TRICKS" COLUMN NIAGARA BUG BULLETIN). BOTH "FIXED" STRINGS AND AN "AREA" ARE HANDLED THE SAME WAY WHEN USING MULTIPLE ARRAYS.

AS I STATED EARLIER, THERE ARE 5 WAYS! THEY ARE:

- 1 - THE HARDWARE METHOD
- 2 - THE MATH METHOD
- 3 - THE BIT-SPLIT METHOD
- 4 - THE DIRECTORY METHOD
- 5 - THE PHANTOM-START METHOD

LET'S TAKE EACH OF THESE IN TURN.

THE HARDWARE METHOD

LAST MONTH WE LOOKED BRIEFLY AT **PLAYER PIANO** (AB HANDBOOK PG.78). @ (A) (LINE 90) = @ (C) (LINE 140) BECAUSE BOTH "A" + "C" ARE ONLY VARIABLES REFERING TO THE @ (X) STRING, AND @ (0) = @ (0) WHETHER THE "0" IS REPRESENTED BY AN "A", OR A "Z"! (UNLESS THE "PHANTOM-START METHOD" IS USED. WE WILL DISCUSS THAT LATER). IF YOU WISHED THERE TO BE ANOTHER @ (0), THE MAKERS OF ASTRO BASIC GAVE US 5 "BUILT IN" (HARDWARE) STRING ALLOCATIONS. THEY ARE:

- @ (X) - $SZ \div 2 - \$(X)$ = NUMBER OF PLACES (WORDS) AVAILABLE
- \$(X) - $SZ \div 2 - @ (X)$ = NUMBER OF PLACES (WORDS) AVAILABLE
- % (-X) - ANYWHERE IN THE TEXT AREA, LIMIT $902 - TEXT \div 2 - @ (X) - \(X)
- % (+X) - UNDER & (10) IN SCREEN, LIMIT $1808 - TEXT$ OR PICTURE SPACE
- % (X) - STACK AREA, SAME AS % (+X), LIMIT $102 - LONGEST LINE \div 2$

THE @(X) STRING AND THE *(X) ARRAY OPERATE THE SAME, OTHER THAN TAPING. FOR EXAMPLE; TO TAPE 150 STRING ENTRIES USE:

```
:PRINT @(0),150 - OR - :PRINT *(149),150
```

ONLY THE *(X) ARRAY TAPES BACKWARDS, BECAUSE IT RESIDES IN MEMORY BACKWARDS. (THERE IS NO BACKWARDS TAPING ROUTINE IN ANY OF THE PRESENT SOFTWARE, ANYWHERE!) %(-X) INCLUDES BOTH REM STATEMENTS AND FREE TEXT SPACE. (SEE PG. 70 AB HANDBOOK AND THE ARCADIAN VOL. 4 PG. 10). THIS SHOWS THE SAME PROGRAM USING %(-X) TWO DIFFERENT WAYS, AS REM STATEMENTS, AND AS FREE SPACE POKES. NOTE: THE PROGRAM ON PG. 10 VOL. 4 ARCADIAN IS FOR BB. FOR AB; %(20050) MUST READ %(20000), ":RETURN;" HAS TO BE DELETED, AND LINES 15 AND 16 (USED FOR TAPING) MUST BE:

```
15 PRINT ";GOTO 6";:PRINT %(16384),1890
```

THE %(+X) AND *(X) ALLOCATIONS REFER TO SCREEN ADDRESSES, WITH %(+X) FROM %(16384) TO %(19999), AND *(X) FROM %(20000) TO %(20358), (SCRATCHPAD). THE END OF THE LINE INPUT BUFFER AND THE "STACK AREA" (DOWN TO %(20358)) ARE THE BEST USEABLE POKES. REMEMBER THOUGH, AB USES THE LINE INPUT BUFFER WHILE RUNNING A PROGRAM, SO THE LONGEST LINE (OF BASIC TEXT) MUST BE TAKEN AS THE STARTING POINT. ALSO, BY NESTING SUBROUTINES AND LOOPS, THE STACK ITSELF GETS DEEPER. %(20358) IS HALF WAY DOWN. "REPACK" STOPS AT THIS ADDRESS ALSO, (IF YOU WISHED TO USE THE UTILITY ON A PROGRAM WITH A FIXED ARRAY). GOING FURTHER BEGS FOR TROUBLE. THE SCREEN ITSELF %(+X) CAN BE USED IN TWO WAYS, BY HIDING IT AT THE BOTTOM UNDER &(10) (&(10) MUST BE BROUGHT UP), OR AT THE TOP UNDER AN INTERRUPT. THERE ARE SOME CARTRIDGES THAT DO THIS, BUT NO AB PROGRAM THAT I HAVE SEEN (OR HEARD OF) YET HAS IT IN. &(9) "HIDES" THE SCRATCHPAD IN MUNCHER, BUT NOBODY USES "HIDE AND SURPRISE" TRICKS TO HIDE STRING DATA IN BASIC. HOWEVER, THE BOTTOM OF THE SCREEN (UNDER &(10)) IS QUITE POPULAR. CANDY MAN [L+M], AND CHICKEN, [BIT FIDDLERS], ARE TWO EXAMPLES. THE TITLE PROGRAM HIDES CANDY MAN, BUT YOU CAN SEE THE DATA WHEN CHICKEN LOADS FROM TAPE. TWO WORDS OF CAUTION ABOUT USING "ON SCREEN" POKES. IF THE SCREEN GETS CLEARED (CLEAR COMMAND), OR SCROLLS, OR ANY GRAPHICS (BOX, LINE, PRINT, TV=, CHRDIS, ETC.) REACH THE STRING AREA, YOU'VE LOST IT! ALSO, BE CAREFULL NOT TO POKE INTO A PLACE WHERE @(X), *(X), %(-X), OR THE BASIC TEXT, WILL RESIDE BENEATH. TO SEE @(X), *(X), %(-X), AND THE TEXT, USE:

```
&(0)=0;&(1)=90;&(2)=132;&(3)=6;&(9)=0  
&(9)=63 RETURNS TO NORMAL.
```

ONE LAST TIP ON "PEEKING" OR "POKING" STRINGS WITH A "FOR NEXT" LOOP, ALWAYS USE "STEP 2", AS BOTH "PEEKING" AND "POKING" ACCESS 2 BYTES (ONE WORD) EACH TIME. NEXT MONTH, MULTIPLE ARRAYS BY SOFTWARE METHODS (RUNNING ALL AT ONCE)! KEEP BUGGIN'!! *