M. David Johnson
https://www.bds-soft.com
info@bds-soft.com

Multi-Platform Programming
And
My Hybrid Quiz Program Process

by M. David Johnson
2022/04/30

Abstract

A system is presented for the development and implementation of Hybrid Multi-Platform Quiz Games.

——

This paper and its associated code are available online at:

http://www.bds-soft.com/cocoPapers.php .

=====

Table of Contents

=====

Introduction

I am *not* a “purist”: I don’t follow the dictum that all Color Computer Programs should be developed and written entirely on a CoCo.

In fact, I hold that whatever works best… well… er… um… works best!

This paper is about my hybrid quiz program process for Multi-Platform Programming This is not the same as “Cross-Platform” programming.

PC Magazine writes:

> Cross platform is a major issue for software developers who want to sell to users no matter which desktop platform they run (Windows, Mac or Linux). There are two primary methods for developing such programs. The first is to compile an executable program into the operating environment (machine language and OS) of each target computer. The second is to use an intermediate language and compile only once…
>
> **Compile to Machine Language**
> The least desired method for developers is to maintain separate sets of source code for the same application; however, it is done routinely when the machine platforms are diverse. For example, C++ applications are compiled directly to the machine language of the target computer. If C++ is used to write a program for Windows and Mac, two separate sets of source code are typically used; one for Windows, one for Mac…
>
> **Compile to an Intermediate Language**
> The second method uses an interpreter such as a Java Virtual Machine. Java is cross platform because a program's source code is compiled into an intermediate "bytecode" language. The bytecode is then executed by a Java Virtual Machine (Java interpreter) that was written for that particular hardware platform.

Conversely, my concept of Multi-Platform (as vs. “Cross-Platform) Programming, would include:

1. Write all the code (for all the items listed below) using TextPad 8 on a Windows 10 machine; and then transfer that code to the various other types of machines for actual running.

2. In Color Computer BASIC, write a series of Data Statements to encompass a given quiz program’s questions, multiple choice answers, etc. (These data statements would vary from quiz-to-quiz).

3. Also in Color Computer BASIC, write a skeleton quiz program which will accept the inclusion of the data statements from item 2, and thus result in a complete Quiz Program for the 64K CoCo 2. (The skeleton would be expected to remain relatively unchanged from quiz program to quiz program).

4. Write a collection of Awk scripts which would run on a Debian 11 Linux server to translate the Color Computer BASIC Data Statements from item 2 into JavaScript Array Assignments.

5. Write a JavaScript skeleton quiz program which will accept the inclusion of the array assignments from item 4, and thus result in a complete Quiz Program for the Games Section of my bds-soft.com website.

6. Write another collection of Awk scripts which would run on the Debian 11 Linux server to translate the Color Computer BASIC Data Statements from item 2 into Python Array Assignments.

7. Write a Python skeleton quiz program which will accept the inclusion of the array assignments from item 6, and thus result in a complete Quiz Program for the Raspberry Pi and other Linux systems.

This process is currently under development, but not yet complete. Steps 1, 2, and 3 are presently well defined, and several resulting 64K CoCo 2 quiz programs are now complete (See https://www.bds-soft.com/php/coco/CoCoBASICGamesPuzzles.php).

The translation steps 4 and 6 are set-up and functioning, at least in a preliminary fashion, and are described in more detail herein.

Steps 5 and 7 remain yet to be addressed.

———

In works of this complexity (at least for me) typos and other errors are bound to sneak in. Please let me know about any you discover so I can note and correct them.

M.D.J. 2022/04/30
info@bds-soft.com

=====

Chapter 01: The CoCo Skeleton

Although the CoCo quiz game skeleton can be expected to vary somewhat for particular game idiosyncrasies, it can also be expected to remain uniform to a high degree.

As an example, the CoCo BASIC skeleton for the Addition Quiz Game is presented here in full.

———

The BASIC program code:

```
00100 '*****
00101 '*
00102 '* ADDITION.BAS
00103 '* MDJ 2022/01/08
00104 '*
00105 '* A QUIZ TO TEST
00106 '* KNOWLEDGE OF
00107 '* ADDITION
00108 '*
00109 '*****

00110 'GO TO THE PROGRAM'S
00120 'EXECUTIVE ROUTINE
00130 GOTO 4220

00131 ' HEADER 00100
00132 ' SELECT 00200
00133 ' EXEC   04000
00134 ' MAIN   05000
00135 ' DATA   10500
00136 ' INIT   27500
00137 ' SPLASH 29500

00200 '*****
00210 '*
00220 '* SELECTION SUBROUTINE:
00230 '*
00240 '* THIS ROUTINE IS UNIFORM
00250 '* AND REMAINS THE SAME FOR
00260 '* EACH AND EVERY QUIZ GAME
00270 '*
00280 '*****

00290 'RANDOM SELECTION OF
```

```
00300 'FOUR OF FIVE.
00310 ' SELECTS FOUR ENTRIES
00320 ' FROM FIVE POSSIBILITIES.
00330 '  INPUTS:
00340 '   NONE
00350 '  OUTPUTS (GLOBAL)
00360 '   RR(4) = THE SELECTED FOUR
00370 '   RP    = POSITION OF "0" ENTRY
00380 '             (0, 1, 2, OR 3)
00390 '         = 4 IF "0" ENTRY
00400 '             NOT INCLUDED

00490 ' INITIALIZATION
00500 FOR I = 0 TO 4
00510   R1(I) = I
00520 NEXT I
00530 RP = 4

00540 'FIRST PASS
00550 R = RND(5) - 1
00560 J = -1
00570 FOR I = 0 TO 4
00580   IF R = I GOTO 610
00590   J = J + 1
00600   R2(J) = R1(I)
00610 NEXT I

00620 'SCRAMBLE PASS
00630 J = RND(24) - 1
00640 FOR I = 0 TO 3
00650   RR(I) = R2(S(J,I))
00660 NEXT I

00670 'POSITION PASS
00680 FOR I = 0 TO 3
00690   IF RR(I) <> 0 GOTO 710
00700   RP = I
00710 NEXT I

00720 RETURN

00730 '*****
00740 '*
00750 '* ENDSUB
00760 '*
00770 '*****
```

```
04000 '*****
04010 '*
04020 '* EXECUTIVE
04030 '*
04040 '*****

04050 'VARIABLES LIST
04060 ' I, J = INDEXES AND COUNTERS
04070 ' A = RESPONSE CODE
04080 ' A$ = KEY PRESS
04090 ' N = NUMBER OF QUESTIONS
04100 ' Q$ = QUESTION
04110 ' QA$ = QUESTION ANSWERS ARRAY
04120 ' QD$ = QUESTIONS DATA ARRAY
04130 ' QN  = QUESTION NUMBER
04140 ' QT  = # OF QUESTIONS ASKED
04150 ' QC  = # OF CORRECT ANSWERS
04160 ' QP  = % CORRECT
04170 ' R, R1, R2, RR, RP
04180 '   = RANDOMIZATION VARIABLES
04190 ' S = SCRAMBLE CODES ARRAY
04200 ' SD$ = SCREEN DISPLAY TEXT

04210 'GO DISPLAY SPLASH SCREEN
04220 GOSUB 29600

04230 'GO INITIALIZE THE SYSTEM
04240 GOSUB 27600

04250 'GO LOAD QUESTIONS DATA
04260 GOSUB 10600

04270 'GO ENTER THE MAIN ROUTINE
04280 GOTO 5110

04290 '*****
04300 '*
04310 '* END EXECUTIVE
04320 '*
04330 '*****

05000 '*****
05010 '*
05020 '* MAIN
05030 '*
05040 '*****
```

```
05050 ' QN  = QUESTION NUMBER
05060 ' Q$ = QUESTION
05070 ' QA$ = QUESTION ANSWERS ARRAY
05080 ' QT  = # OF QUESTIONS ASKED
05090 ' QC  = # OF CORRECT ANSWERS
05100 ' QP  = % CORRECT
05110 DIM QA$(4)
05120 QT = 0
05130 QC = 0
05140 QP = 0

05150 'RANDOMLY SELECT A
05160 'QUESTION
05170 'ADJUST RND FOR TOTAL
05180 'NUMBER OF QUESTIONS
05190 QN = RND(90) - 1
05200 Q$ = QD$(QN,0)

05210 'RANDOMLY SELECT AND
05220 'SCRAMBLE FOUR OF
05230 'THE FIVE ANSWERS
05240 GOSUB 500
05250 FOR I = 0 TO 3
05260   QA$(I) = QD$(QN,RR(I)+1)
05270 NEXT I

05280 'MAIN DISPLAY SCREEN
05290 PRINT@32,"  WHAT IS THE RESULT:"
05300 PRINT@64,"  "+Q$+" ?"
05310 PRINT@128, "      A. "+QA$(0)
05320 PRINT@160, "      B. "+QA$(1)
05330 PRINT@192, "      C. "+QA$(2)
05340 PRINT@224, "      D. "+QA$(3)
05350 PRINT@256, "      E. NONE OF THE ABOVE"
05360 PRINT@320, "  PRESS A, B, C, D, OR E"
05370 PRINT@352, "                                    "
05380 PRINT@384, "                                    "
05390 PRINT@416, "                                    "
05400 PRINT@448, "  SCORE = ";QC;" OF ";QT;" =
";QP;"%";

05410 A$ = INKEY$
05420 IF A$="" GOTO 5410

05430 IF (A$="A" OR A$=CHR$(97)) GOTO 5520
05440 IF (A$="B" OR A$=CHR$(98)) GOTO 5530
05450 IF (A$="C" OR A$=CHR$(99)) GOTO 5540
```

```
05460 IF (A$="D" OR A$=CHR$(100)) GOTO 5550
05470 IF (A$="E" OR A$=CHR$(101)) GOTO 5560
05480 SOUND 159,2
05490 SOUND 159,2
05500 PRINT@320, "  YOU MUST PRESS ONE OF A TO E"
05510 GOTO 5410

05520 A = 0: GOTO 5570
05530 A = 1: GOTO 5570
05540 A = 2: GOTO 5570
05550 A = 3: GOTO 5570
05560 A = 4: GOTO 5570

05570 IF A = RP GOTO 5710
05580 SOUND 78,1
05590 SOUND 78,1
05600 SOUND 78,1
05610 SOUND 5,3
05620 PRINT@320, "    **SORRY: THAT IS INCORRECT."
05630 SD$ = "  CORRECT = "
05640 IF RP=0 THEN SD$ = SD$ + "A. "+QA$(0)
05650 IF RP=1 THEN SD$ = SD$ + "B. "+QA$(1)
05660 IF RP=2 THEN SD$ = SD$ + "C. "+QA$(2)
05670 IF RP=3 THEN SD$ = SD$ + "D. "+QA$(3)
05680 IF RP=4 THEN SD$ = SD$ + "E.NONE OF THE ABOVE"
05690 PRINT@352, SD$
05700 GOTO 5770

05710 SOUND 204,1
05720 SOUND 204,1
05730 SOUND 204,1
05740 SOUND 218,3
05750 PRINT@320, "    *** THAT IS CORRECT! ***"
05760 QC = QC + 1

05770 QT = QT + 1
05780 QP = INT(100 * (QC/QT))
05790 PRINT@448, "  SCORE = ";QC;" OF ";QT;" =
";QP;"%";
05800 PRINT@384, "  PRESS C TO CONTINUE;Q TO QUIT"

05810 A$ = INKEY$
05820 IF A$="" GOTO 5810

05830 IF (A$="C" OR A$=CHR$(99)) GOTO 5190
05840 IF (A$="Q" OR A$=CHR$(113)) GOTO 5890
05850 SOUND 159,2
```

```
05860 SOUND 159,2
05870 PRINT@384, "  YOU MUST PRESS C OR Q"
05880 GOTO 5810

05890 CLS
05900 PRINT:PRINT "   FINAL SCORE = ";QP;"%"
05910 PRINT:PRINT "   THANKS FOR PLAYING!"
05920 PRINT "     COME BACK SOON."
05930 PRINT:PRINT "   M. DAVID JOHNSON"
05940 PRINT "   INFO@BDS-SOFT.COM"
05950 PRINT:PRINT:PRINT:PRINT:PRINT

05960 GOTO 32767

05970 '*****
05980 '*
05990 '* END MAIN
06000 '*
06010 '*****

10500 '*****
10510 '*
10520 '* STANDARD BASIC QUIZ GAMES
10530 '* DATA LOADING ROUTINE
10540 '*
10550 '*****

10560 'THIS DATA MUST BE
10570 'INDIVIDUALIZED FOR
10580 'EACH SPECIFIC QUIZ

10590 'N = NUMBER OF QUESTIONS
10600 N = 90

10610 'QD$ = QUESTIONS DATA ARRAY
10620 DIM QD$(N,6)
10630 FOR I = 0 TO N-1
10640   FOR J = 0 TO 5
10650     READ QD$(I,J)
10660   NEXT J
10670 NEXT I

10680 PRINT "  OKAY -"
10690 PRINT "  ALL LOADED AND READY TO GO"
10700 PRINT
10710 PRINT "  PRESS ANY KEY TO BEGIN"
10720 PRINT
```

```
10730 A$=INKEY$
10740 IF A$="" GOTO 10730

10750 RETURN

10760 '*****
10770 '*
10780 '* END DATA LOADING
10790 '*
10800 '*****

10810 ‘ DATA LIST GOES HERE

27500 '*****
27510 '*
27520 '* STANDARD BASIC QUIZ GAMES
27530 '* INITIALIZATION ROUTINE
27540 '*
27550 '*****

27600 CLS

27610 PRINT
27620 PRINT "  LOADING THE SYSTEM"
27630 PRINT "  PLEASE WAIT"
27640 PRINT

27650 'RANDOMIZATION ARRAYS
27660 DIM R1(5)
27670 DIM R2(4)
27680 DIM RR(4)

27690 'SCRAMBLE CODES ARRAY
27700 DIM S(24,4)

27710 'RANDOMIZE THE RND FUNCTION
27720 R = RND(-TIMER)

27800 'INITIALIZE THE SCRAMBLE CODES

27810 S(0,0) = 0
27820 S(0,1) = 1
27830 S(0,2) = 2
27840 S(0,3) = 3

27850 S(1,0) = 0
```

```
27860 S(1,1) = 1
27870 S(1,2) = 3
27880 S(1,3) = 2

27890 S(2,0) = 0
27900 S(2,1) = 2
27910 S(2,2) = 1
27920 S(2,3) = 3

27930 S(3,0) = 0
27940 S(3,1) = 2
27950 S(3,2) = 3
27960 S(3,3) = 1

27970 S(4,0) = 0
27980 S(4,1) = 3
27990 S(4,2) = 1
28000 S(4,3) = 2

28010 S(5,0) = 0
28020 S(5,1) = 3
28030 S(5,2) = 2
28040 S(5,3) = 1

28050 S(6,0) = 1
28060 S(6,1) = 0
28070 S(6,2) = 2
28080 S(6,3) = 3

28090 S(7,0) = 1
28100 S(7,1) = 0
28110 S(7,2) = 3
28120 S(7,3) = 2

28130 S(8,0) = 1
28140 S(8,1) = 2
28150 S(8,2) = 0
28160 S(8,3) = 3

28170 S(9,0) = 1
28180 S(9,1) = 2
28190 S(9,2) = 3
28200 S(9,3) = 0

28210 S(10,0) = 1
28220 S(10,1) = 3
28230 S(10,2) = 0
```

```
28240 S(10,3) = 2

28250 S(11,0) = 1
28260 S(11,1) = 3
28270 S(11,2) = 2
28280 S(11,3) = 0

28290 S(12,0) = 2
28300 S(12,1) = 0
28310 S(12,2) = 1
28320 S(12,3) = 3

28330 S(13,0) = 2
28340 S(13,1) = 0
28350 S(13,2) = 3
28360 S(13,3) = 1

28370 S(14,0) = 2
28380 S(14,1) = 1
28390 S(14,2) = 0
28400 S(14,3) = 3

28410 S(15,0) = 2
28420 S(15,1) = 1
28430 S(15,2) = 3
28440 S(15,3) = 0

28450 S(16,0) = 2
28460 S(16,1) = 3
28470 S(16,2) = 0
28480 S(16,3) = 1

28490 S(17,0) = 2
28500 S(17,1) = 3
28510 S(17,2) = 1
28520 S(17,3) = 0

28530 S(18,0) = 3
28540 S(18,1) = 0
28550 S(18,2) = 1
28560 S(18,3) = 2

28570 S(19,0) = 3
28580 S(19,1) = 0
28590 S(19,2) = 2
28600 S(19,3) = 1
```

```
28610 S(20,0) = 3
28620 S(20,1) = 1
28630 S(20,2) = 0
28640 S(20,3) = 2

28650 S(21,0) = 3
28660 S(21,1) = 1
28670 S(21,2) = 2
28680 S(21,3) = 0

28690 S(22,0) = 3
28700 S(22,1) = 2
28710 S(22,2) = 0
28720 S(22,3) = 1

28730 S(23,0) = 3
28740 S(23,1) = 2
28750 S(23,2) = 1
28760 S(23,3) = 0

28770 RETURN

28780 '*****
28790 '*
28800 '* END INITIALIZATION
28810 '*
28820 '*****

29500 '*****
29510 '*
29520 '* DISPLAY THE STANDARD
29530 '* BASIC QUIZ GAMES
29540 '* SPLASH SCREEN
29550 '*
29560 '*****

29600 CLS

29610 'LINES 0 AND 1 ARE BLANK GREEN

29620 'DISPLAY UNIFORM STANDARD LINE 2
29630 SD$ = CHR$(32) + CHR$(158)
29640 FOR I = 0 TO 27
29650   SD$ = SD$ + CHR$(156)
29660 NEXT I
29670 SD$ = SD$ + CHR$(157) + CHR$(32)
29680 PRINT@64, SD$
```

```
29690 'DISPLAY UNIFORM STANDARD LINE 3
29700 SD$ = CHR$(32) + CHR$(154) + CHR$(206)
29710 FOR I = 0 TO 25
29720   SD$ = SD$ + CHR$(204)
29730 NEXT I
29740 SD$ = SD$ + CHR$(205) + CHR$(149) + CHR$(32)
29750 PRINT@96, SD$

29760 'DISPLAY UNIFORM STANDARD LINE 4
29770 SD$ = CHR$(32) + CHR$(154) + CHR$(202)
29780 FOR I = 0 TO 25
29790   SD$ = SD$ + CHR$(32)
29800 NEXT I
29810 SD$ = SD$ + CHR$(197) + CHR$(149) + CHR$(32)
29820 PRINT@128, SD$

29830 'DISPLAY SPECIFIC STANDARD LINE 5
29840 SD$ = CHR$(32) + CHR$(154) + CHR$(202)
29850 ' *** CHANGE NEXT LINE AS REQUIRED:
29860 SD$ = SD$ + "    ADDITION  PRACTICE    "
29870 SD$ = SD$ + CHR$(197) + CHR$(149) + CHR$(32)
29880 PRINT@160, SD$

29890 'DISPLAY SPECIFIC STANDARD LINE 6
29900 SD$ = CHR$(32) + CHR$(154) + CHR$(202)
29910 ' *** CHANGE NEXT LINE AS REQUIRED:
29920 SD$ = SD$ + "        QUIZ GAME         "
29930 SD$ = SD$ + CHR$(197) + CHR$(149) + CHR$(32)
29940 PRINT@192, SD$

29950 'DISPLAY UNIFORM STANDARD LINE 7
29960 SD$ = CHR$(32) + CHR$(154) + CHR$(202)
29970 FOR I = 0 TO 25
29980   SD$ = SD$ + CHR$(32)
29990 NEXT I
30000 SD$ = SD$ + CHR$(197) + CHR$(149) + CHR$(32)
30010 PRINT@224, SD$

30020 'DISPLAY UNIFORM STANDARD LINE 8
30030 SD$ = CHR$(32) + CHR$(154) + CHR$(203)
30040 FOR I = 0 TO 25
30050   SD$ = SD$ + CHR$(195)
30060 NEXT I
30070 SD$ = SD$ + CHR$(199) + CHR$(149) + CHR$(32)
30080 PRINT@256, SD$
```

```
30090 'DISPLAY UNIFORM STANDARD LINE 9
30100 SD$ = CHR$(32) + CHR$(155)
30110 FOR I = 0 TO 27
30120   SD$ = SD$ + CHR$(147)
30130 NEXT I
30140 SD$ = SD$ + CHR$(151) + CHR$(32)
30150 PRINT@288, SD$

30160 'LINE 10 IS BLANK GREEN

30170 'DISPLAY UNIFORM STANDARD LINE 11
30180 PRINT@352, "   PRESS ANY KEY  TO CONTINUE   "

30190 'LINES 12 AND 13 ARE BLANK GREEN

30200 'DISPLAY SPECIFIC STANDARD LINE 14
30210 ' *** CHANGE THE YEAR AS REQUIRED
30220 PRINT@448, "       BDS SOFTWARE  2022       ";

30230 'LINE 15 IS BLANK GREEN

30240 A$ = INKEY$
30250 IF A$="" GOTO 30240

30260 RETURN

30270 '*****
30280 '*
30290 '* END SPLASH SCREEN
30290 '*
30300 '*****

32767 END
```

======

Chapter 02: The CoCo Data

Here is an abbreviated excerpt from the CoCo BASIC Data List for the Addition Quiz.

_____

```
11000 DATA 2 PLUS 2 =
11010 DATA 4
11020 DATA 2
11030 DATA 3
11040 DATA 5
11050 DATA 6

11100 DATA 2 PLUS 3 =
11110 DATA 5
11120 DATA 4
11130 DATA 6
11140 DATA 7
11150 DATA 8

11200 DATA 2 PLUS 4 =
11210 DATA 6
11220 DATA 4
11230 DATA 5
11240 DATA 7
11250 DATA 8
...

19800 DATA 12 PLUS 11 =
19810 DATA 23
19820 DATA 21
19830 DATA 19
19840 DATA 25
19850 DATA 27

19900 DATA 12 PLUS 12 =
19910 DATA 24
19920 DATA 22
19930 DATA 20
19940 DATA 26
19950 DATA 28
```

======

Chapter 03: Universal Step 1

The translation process involves four steps: two universal steps followed by two JavaScript-specific steps; or the same two universal steps followed by two Python-specific steps. Each of these steps consists of one awk script.

This chapter presents the first universal step.

———

The awk script code:

```
#! /bin/awk -f

#####
#
# rp00.awk
# MDJ 2022/03/26
#
# Universal Version Step One:
#   Read and Print All Lines.
#     i.e. Check for gross syntax
#     errors in the file identified
#     on the command line.
#
# To Run:
#   cdCoCo
#   cd wk00
#   rp00.awk filename
#
#####

{
     print $0
}

#####
#
# EOF
#
#####
```

======

Chapter 04: Universal Step 2

The awk script code:

```
#! /bin/awk -f

#####
#
# integ00.awk
# MDJ 2022/03/26
#
# Universal Version Step Two:
#   Check the integrity of the
a   file identified on the
#   command line.
#
# To Run:
#   cdCoCo
#   cd wk00
#   integ00.awk filename
#
#####

BEGIN {
     BNn = 0   # Block Number Counter
     BN = "00" # Block Number Counter Display
     print " "
     print "Beginning of Integrity Test"
     print " "
}

{

     # First Block
     # First Line of Block
     #  = Question Number Comment Line

     # Truncate $NF to remove trailing "\n"
     NFT = substr($NF,1,2)

     # Convert NFT string to numeric
     NFn = 0 + NFT

     # Convert BNn Counter to string
     if ( BNn < 10 )
     {
```

```
     BN = "0" BNn
}
else
{
     BN = "" BNn
}

if ( NFn == BNn )
{
     print "Match " BN " " NFT
     print $0
}
else
{
     print "No Match " BN " " NFT " ERROR!"
     print $0
     exit 1
}

# Question Line
getline
print $0

# Correct Answer Line
getline
print $0

# Incorrect Answer #1 Line
getline
print $0

# Incorrect Answer #2 Line
getline
print $0

# Incorrect Answer #3 Line
getline
print $0

# Incorrect Answer #4 Line
getline
print $0

# Blank Line
getline
print $0
```

```
	# Blocks 2 through last:
	# Question Number Comment Line
	while ( 1 == 1 )
	{
		r = getline
		# If EOF:
		if (r <= 0)
		{
			exit 0
		}
		else
		{
			BNn = BNn + 1

			# Truncate $NF to remove trailing "\n"
			NFT = substr($NF,1,2)

			# Convert NFT string to numeric
			NFn = 0 + NFT

			# Convert BNn Counter to string
			if ( BNn < 10 )
			{
				BN = "0" BNn
			}
			else
			{
				BN = "" BNn
			}

			if ( NFn == BNn )
			{
				print "Match " BN " " NFT
				print $0
			}
			else
			{
				print "No Match " BN " " NFT "
ERROR!"
				print $0
				exit 1
			}
		}

		# Question Line
		getline
		print $0
```

```
                # Correct Answer Line
                getline
                print $0

                # Incorrect Answer #1 Line
                getline
                print $0

                # Incorrect Answer #2 Line
                getline
                print $0

                # Incorrect Answer #3 Line
                getline
                print $0

                # Incorrect Answer #4 Line
                getline
                print $0

                # Blank Line
                getline
                print $0
           }
      }

      END {
           print "End of Integrity Test"
           print " "
      }

      #####
      #
      # EOF
      #
      #####
```

======

Chapter 05: JavaScript Step 3

The awk script code:

```
#! /bin/awk -f

#####
#
# jsCopy00.awk
# MDJ 2022/03/26
#
# JavaScript Version Step Three:
#   Copy the file identified
a   on the command line to
#   wk00/dataout; while also
#   adding the Question Number
#   Comment line.
#
# To Run:
#   cdCoCo
#   cd wk00
#   pyCopy00.awk filename
#
#####

BEGIN {
     BNn = 0   # Block Number Counter
     BN = "00" # Block Number Counter Display
     print " "
     print "Beginning of Step Three Copy"
     print " "
}

{

     # First Block
     # First Line of Block
     #  = Question Number Comment Line

     # Truncate $NF to remove trailing "\n"
     NFT = substr($NF,1,2)

     # Convert NFT string to numeric
     NFn = 0 + NFT

     # Add closing semicolon to string
```

```
NFT = NFT ";"
$NF = NFT

# Convert BNn Counter to string
if ( BNn < 10 )
{
     BN = "0" BNn "\;"
}
else
{
     BN = "" BNn "\;"
}

if ( NFn == BNn )
{
     print "Match " BN " " NFT
     print $0
     print $0 > "../wk00/dataout"
}
else
{
     print "No Match " BN " " NFT " ERROR!"
     print $0
     exit 1
}

# Question Line
getline
print $0
print $0 >> "../wk00/dataout"

# Correct Answer Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #1 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #2 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #3 Line
```

```
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #4 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Blank Line
getline
print $0
print $0 >> "../wk00/dataout"

# Blocks 2 through last:
# Question Number Comment Line
while ( 1 == 1 )
{
     r = getline
     # If EOF:
     if (r <= 0)
     {
          exit 0
     }
     else
     {
          BNn = BNn + 1

          # Truncate $NF to remove trailing "\n"
          NFT = substr($NF,1,2)

          # Convert NFT string to numeric
          NFn = 0 + NFT

          # Add closing semicolon to string
          NFT = NFT ";"
          $NF = NFT

          # Convert BNn Counter to string
          if ( BNn < 10 )
          {
               BN = "0" BNn "\;"
          }
          else
          {
               BN = "" BNn "\;"
          }
```

```
                    if ( NFn == BNn )
                    {
                         print "Match " BN " " NFT
                         print $0
                         print $0 >> "../wk00/dataout"
                    }
                    else
                    {
                         print "No Match " BN " " NFT "
ERROR!"
                         print $0
                         exit 1
                    }
               }

               # Question Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Correct Answer Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Incorrect Answer #1 Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Incorrect Answer #2 Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Incorrect Answer #3 Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Incorrect Answer #4 Line
               getline
               print $0
               print $0 >> "../wk00/dataout"

               # Blank Line
```

```
                getline
                print $0
                print $0 >> "../wk00/dataout"
           }
     }

     END {
           print "End of Step Three Copy"
           print " "
     }

     #####
     #
     # EOF
     #
     #####

======
```

Chapter 06: JavaScript Step 4

The awk script code:

```
#! /bin/awk -f

#####
#
# jsMath00.awk
# MDJ 2022/03/26
#
# Use this program only for:
#   AdditionData00
#   AdvAddData00
#   AdvMultData00
#   AdvSubData00
#   MultiplyData00
#   SubtractData00
#
# JavaScript Version Step Four:
#   Copy wk00/dataout
#   to wk01/dataout
#   while adding the
#   Prequestion Line
#   to each block; and
#   then translating the
#   Question and Answer
#   Lines as well.
#
# To Run:
#   cdCoCo
#   cd wk00
#   jsMath00.awk dataout
#
# After completion of run, copy
#   /wk01/dataout to final file
#     /wk01/AdditionData01,
#     /wk01/AdvAddData01,
#     /wk01/AdvMult01,
#     /wk01/AdvSubData01,
#     /wk01/MultiplyData01,or
#     /wk01/SubtractData01,
#     as appropriate.
#
#####
```

```
BEGIN {
     BNn = 0   # Block Number Counter
     BN = "00" # Block Number Counter Display
     # Prequestion Line:
     PQL = "What is the result:"
     # Array Line Fragments
     ALA = "AStr["
     ALB = "]["
     ALC = "]"
     print " "
     print "Start Step Four jsMath Xfer"
     print " "
}

{

     # First Block
     # First Line of Block
     #  = Question Number Comment Line

     # Truncate $NF to remove trailing "\n"
     NFT = substr($NF,1,2)

     # Convert NFT string to numeric
     NFn = 0 + NFT

     # Convert BNn Counter to string
     if ( BNn < 10 )
     {
          BN = "0" BNn
     }
     else
     {
          BN = "" BNn
     }

     if ( NFn == BNn )
     {
          print "Match " BN " " NFT
          print $0
          print $0 > "../wk01/dataout"
          PQLA = "AStr[" BNn "][0] = \"" PQL "\";"
          print PQLA
          print PQLA >> "../wk01/dataout"
     }
     else
     {
```

```
     print "No Match " BN " " NFT " ERROR!"
     print $0
     exit 1
}

# Question Line
getline
$1 = ALA BNn ALB "1" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     $NF = "\"" $NF " ?\";"
}
else
{
     $3 = "\"" $3
     $NF = $NF " ?\";"
}
print $0
print $0 >> "../wk01/dataout"

# Correct Answer Line
getline
$1 = ALA BNn ALB "2" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #1 Line
getline
$1 = ALA BNn ALB "3" ALC
```

```
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #2 Line
getline
$1 = ALA BNn ALB "4" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #3 Line
getline
$1 = ALA BNn ALB "5" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
```

```
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #4 Line
getline
$1 = ALA BNn ALB "6" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Blank Line
getline
print $0
print $0 >> "../wk01/dataout"

# Blocks 2 through last:
# Question Number Comment Line
while ( 1 == 1 )
{
     r = getline
     # If EOF:
     if (r <= 0)
     {
```

```
                exit 0
        }
        else
        {
                BNn = BNn + 1

                # Truncate $NF to remove trailing "\n"
                NFT = substr($NF,1,2)

                # Convert NFT string to numeric
                NFn = 0 + NFT

                # Convert BNn Counter to string
                if ( BNn < 10 )
                {
                        BN = "0" BNn
                }
                else
                {
                        BN = "" BNn
                }

                if ( NFn == BNn )
                {
                        print "Match " BN " " NFT
                        print $0
                        print $0 >> "../wk01/dataout"
                        PQLA = "AStr[" BNn "][0] = \"" PQL
"\";"
                        print PQLA
                        print PQLA >> "../wk01/dataout"
                }
                else
                {
                        print "No Match " BN " " NFT "
ERROR!"
                        print $0
                        exit 1
                }
        }

        # Question Line
        getline
        $1 = ALA BNn ALB "1" ALC
        $2 = "="
        len = length($NF)
        $NF = substr($NF,1,len-1)
```

```
if (NF == 3)
{
     $NF = "\"" $NF " ?\";"
}
else
{
     $3 = "\"" $3
     $NF = $NF " ?\";"
}
print $0
print $0 >> "../wk01/dataout"

# Correct Answer Line
getline
$1 = ALA BNn ALB "2" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #1 Line
getline
$1 = ALA BNn ALB "3" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
```

```
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #2 Line
getline
$1 = ALA BNn ALB "4" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
print $0 >> "../wk01/dataout"

# Incorrect Answer #3 Line
getline
$1 = ALA BNn ALB "5" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\";"
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\";"
}
print $0
```

```
                print $0 >> "../wk01/dataout"

                # Incorrect Answer #4 Line
                getline
                $1 = ALA BNn ALB "6" ALC
                $2 = "="
                len = length($NF)
                $NF = substr($NF,1,len-1)
                if (NF == 3)
                {
                     if (substr($NF,1,1) != "\"")
                     {
                          $NF = "\"" $NF "\";"
                     }
                }
                else
                {
                     $3 = "\"" $3
                     $NF = $NF "\";"
                }
                print $0
                print $0 >> "../wk01/dataout"

                # Blank Line
                getline
                print $0
                print $0 >> "../wk01/dataout"
           }
      }

      END {
           print "End of Step Four jsMath Xfer"
           print " "
      }

      #####
      #
      # EOF
      #
      #####

======
```

Chapter 07: JavaScript Result

Here is an excerpt from the JavaScript Array Assignments list after the translation is complete:

_____

```
// Question Number 00;
AStr[0][0] = "What is the result:";
AStr[0][1] = "2 PLUS 2 = ?";
AStr[0][2] = "4";
AStr[0][3] = "2";
AStr[0][4] = "3";
AStr[0][5] = "5";
AStr[0][6] = "6";

// Question Number 01;
AStr[1][0] = "What is the result:";
AStr[1][1] = "2 PLUS 3 = ?";
AStr[1][2] = "5";
AStr[1][3] = "4";
AStr[1][4] = "6";
AStr[1][5] = "7";
AStr[1][6] = "8";

// Question Number 02;
AStr[2][0] = "What is the result:";
AStr[2][1] = "2 PLUS 4 = ?";
AStr[2][2] = "6";
AStr[2][3] = "4";
AStr[2][4] = "5";
AStr[2][5] = "7";
AStr[2][6] = "8";

…

// Question Number 88;
AStr[88][0] = "What is the result:";
AStr[88][1] = "12 PLUS 11 = ?";
AStr[88][2] = "23";
AStr[88][3] = "21";
AStr[88][4] = "19";
AStr[88][5] = "25";
AStr[88][6] = "27";

// Question Number 89;
AStr[89][0] = "What is the result:";
```

```
     AStr[89][1] = "12 PLUS 12 = ?";
     AStr[89][2] = "24";
     AStr[89][3] = "22";
     AStr[89][4] = "20";
     AStr[89][5] = "26";
     AStr[89][6] = "28";
```

======

Chapter 08: Python Step 3

The awk script code:

```
#! /bin/awk -f

#####
#
# pyCopy00.awk
# MDJ 2022/03/27
#
# Python Version Step Three:
#   Copy the file identified
a   on the command line to
#   wk00/dataout; while also
#   adding the Question Number
#   Comment line.
#
# To Run:
#   cdCoCo
#   cd wk00
#   pyCopy00.awk filename
#
#####

BEGIN {
     BNn = 0   # Block Number Counter
     BN = "00" # Block Number Counter Display
     print " "
     print "Beginning of Step Three Copy"
     print " "
}

{

     # First Block
     # First Line of Block
     #  = Question Number Comment Line

     # Replace the leading "//" with "#"
     len = length($1)
     t1 = substr($1,3,len)
     $1 = "#" t1

     # Truncate $NF to remove trailing "\n"
     NFT = substr($NF,1,2)
```

```
# Convert NFT string to numeric
NFn = 0 + NFT

# Convert BNn Counter to string
if ( BNn < 10 )
{
     BN = "0" BNn
}
else
{
     BN = "" BNn
}

if ( NFn == BNn )
{
     print "Match " BN " " NFT
     print $0
     print $0 > "../wk00/dataout"
}
else
{
     print "No Match " BN " " NFT " ERROR!"
     print $0
     exit 1
}

# Question Line
getline
print $0
print $0 >> "../wk00/dataout"

# Correct Answer Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #1 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #2 Line
getline
print $0
print $0 >> "../wk00/dataout"
```

```
# Incorrect Answer #3 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Incorrect Answer #4 Line
getline
print $0
print $0 >> "../wk00/dataout"

# Blank Line
getline
print $0
print $0 >> "../wk00/dataout"

# Blocks 2 through last:
# Question Number Comment Line
while ( 1 == 1 )
{
     r = getline
     # If EOF:
     if (r <= 0)
     {
          exit 0
     }
     else
     {
          BNn = BNn + 1

          # Replace the leading "//" with "#"
          len = length($1)
          t1 = substr($1,3,len)
          $1 = "#" t1

          # Truncate $NF to remove trailing "\n"
          NFT = substr($NF,1,2)

          # Convert NFT string to numeric
          NFn = 0 + NFT

          # Convert BNn Counter to string
          if ( BNn < 10 )
          {
               BN = "0" BNn
          }
          else
          {
```

```
                    BN = "" BNn
               }

               if ( NFn == BNn )
               {
                    print "Match " BN " " NFT
                    print $0
                    print $0 >> "../wk00/dataout"
               }
               else
               {
                    print "No Match " BN " " NFT "
ERROR!"
                    print $0
                    exit 1
               }
          }

          # Question Line
          getline
          print $0
          print $0 >> "../wk00/dataout"

          # Correct Answer Line
          getline
          print $0
          print $0 >> "../wk00/dataout"

          # Incorrect Answer #1 Line
          getline
          print $0
          print $0 >> "../wk00/dataout"

          # Incorrect Answer #2 Line
          getline
          print $0
          print $0 >> "../wk00/dataout"

          # Incorrect Answer #3 Line
          getline
          print $0
          print $0 >> "../wk00/dataout"

          # Incorrect Answer #4 Line
          getline
          print $0
          print $0 >> "../wk00/dataout"
```

```
                # Blank Line
                getline
                print $0
                print $0 >> "../wk00/dataout"
           }
      }

      END {
           print "End of Step Three Copy"
           print " "
      }

      #####
      #
      # EOF
      #
      #####

======
```

Chapter 09: Python Step 4

The awk script code:

```
#! /bin/awk -f

#####
#
# pyMath00.awk
# MDJ 2022/03/27
#
# Use this program only for:
#   AdditionData00
#   AdvAddData00
#   AdvMultData00
#   AdvSubData00
#   MultiplyData00
#   SubtractData00
#
# Python Version Step Four:
#   Copy wk00/dataout
#   to wk02/dataout
#   while adding the
#   Prequestion Line
#   to each block; and
#   then translating the
#   Question and Answer
#   Lines as well.
#
# To Run:
#   cdCoCo
#   cd wk00
#   pyMath00.awk dataout
#
# After completion of run, copy
#   /wk02/dataout to final file
#     /wk02/AdditionData02,
#     /wk02/AdvAddData02,
#     /wk02/AdvMult02,
#     /wk02/AdvSubData02,
#     /wk02/MultiplyData02,or
#     /wk02/SubtractData02,
#     as appropriate.
#
#####
```

```
BEGIN {
     BNn = 0   # Block Number Counter
     BN = "00" # Block Number Counter Display
     # Prequestion Line:
     PQL = "What is the result:"
     # Array Line Fragments
     ALA = "AStr["
     ALB = "]["
     ALC = "]"
     print " "
     print "Start Step Four pyMath Xfer"
     print " "
}

{

     # First Block
     # First Line of Block
     #  = Question Number Comment Line

     # Truncate $NF to remove trailing "\n"
     NFT = substr($NF,1,2)

     # Convert NFT string to numeric
     NFn = 0 + NFT

     # Convert BNn Counter to string
     if ( BNn < 10 )
     {
          BN = "0" BNn
     }
     else
     {
          BN = "" BNn
     }

     if ( NFn == BNn )
     {
          print "Match " BN " " NFT
          print $0
          print $0 > "../wk02/dataout"
          PQLA = "AStr[" BNn "][0] = \"" PQL "\""
          print PQLA
          print PQLA >> "../wk02/dataout"
     }
     else
     {
```

```
     print "No Match " BN " " NFT " ERROR!"
     print $0
     exit 1
}

# Question Line
getline
$1 = ALA BNn ALB "1" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     $NF = "\"" $NF " ?\""
}
else
{
     $3 = "\"" $3
     $NF = $NF " ?\""
}
print $0
print $0 >> "../wk02/dataout"

# Correct Answer Line
getline
$1 = ALA BNn ALB "2" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #1 Line
getline
$1 = ALA BNn ALB "3" ALC
```

```
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #2 Line
getline
$1 = ALA BNn ALB "4" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #3 Line
getline
$1 = ALA BNn ALB "5" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
```

```
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #4 Line
getline
$1 = ALA BNn ALB "6" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Blank Line
getline
print $0
print $0 >> "../wk02/dataout"

# Blocks 2 through last:
# Question Number Comment Line
while ( 1 == 1 )
{
     r = getline
     # If EOF:
     if (r <= 0)
     {
```

```
                exit 0
        }
        else
        {
                BNn = BNn + 1

                # Truncate $NF to remove trailing "\n"
                NFT = substr($NF,1,2)

                # Convert NFT string to numeric
                NFn = 0 + NFT

                # Convert BNn Counter to string
                if ( BNn < 10 )
                {
                        BN = "0" BNn
                }
                else
                {
                        BN = "" BNn
                }

                if ( NFn == BNn )
                {
                        print "Match " BN " " NFT
                        print $0
                        print $0 >> "../wk02/dataout"
                        PQLA = "AStr[" BNn "][0] = \"" PQL
"\""
                        print PQLA
                        print PQLA >> "../wk02/dataout"
                }
                else
                {
                        print "No Match " BN " " NFT "
ERROR!"
                        print $0
                        exit 1
                }
        }

        # Question Line
        getline
        $1 = ALA BNn ALB "1" ALC
        $2 = "="
        len = length($NF)
        $NF = substr($NF,1,len-1)
```

```
if (NF == 3)
{
     $NF = "\"" $NF " ?\""
}
else
{
     $3 = "\"" $3
     $NF = $NF " ?\""
}
print $0
print $0 >> "../wk02/dataout"

# Correct Answer Line
getline
$1 = ALA BNn ALB "2" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #1 Line
getline
$1 = ALA BNn ALB "3" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
```

```
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #2 Line
getline
$1 = ALA BNn ALB "4" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
print $0 >> "../wk02/dataout"

# Incorrect Answer #3 Line
getline
$1 = ALA BNn ALB "5" ALC
$2 = "="
len = length($NF)
$NF = substr($NF,1,len-1)
if (NF == 3)
{
     if (substr($NF,1,1) != "\"")
     {
          $NF = "\"" $NF "\""
     }
}
else
{
     $3 = "\"" $3
     $NF = $NF "\""
}
print $0
```

```
                print $0 >> "../wk02/dataout"

                # Incorrect Answer #4 Line
                getline
                $1 = ALA BNn ALB "6" ALC
                $2 = "="
                len = length($NF)
                $NF = substr($NF,1,len-1)
                if (NF == 3)
                {
                     if (substr($NF,1,1) != "\"")
                     {
                          $NF = "\"" $NF "\""
                     }
                }
                else
                {
                     $3 = "\"" $3
                     $NF = $NF "\""
                }
                print $0
                print $0 >> "../wk02/dataout"

                # Blank Line
                getline
                print $0
                print $0 >> "../wk02/dataout"
           }
      }

      END {
           print "End of Step Four pyMath Xfer"
           print " "
      }

      #####
      #
      # EOF
      #
      #####

======
```

Chapter 10: Python Result

Here is an excerpt from the Python Array Assignments list after the translation is complete:

_____

```
# Question Number 00
AStr[0][0] = "What is the result:"
AStr[0][1] = "2 PLUS 2 = ?"
AStr[0][2] = "4"
AStr[0][3] = "2"
AStr[0][4] = "3"
AStr[0][5] = "5"
AStr[0][6] = "6"

# Question Number 01
AStr[1][0] = "What is the result:"
AStr[1][1] = "2 PLUS 3 = ?"
AStr[1][2] = "5"
AStr[1][3] = "4"
AStr[1][4] = "6"
AStr[1][5] = "7"
AStr[1][6] = "8"

# Question Number 02
AStr[2][0] = "What is the result:"
AStr[2][1] = "2 PLUS 4 = ?"
AStr[2][2] = "6"
AStr[2][3] = "4"
AStr[2][4] = "5"
AStr[2][5] = "7"
AStr[2][6] = "8"

…

# Question Number 88
AStr[88][0] = "What is the result:"
AStr[88][1] = "12 PLUS 11 = ?"
AStr[88][2] = "23"
AStr[88][3] = "21"
AStr[88][4] = "19"
AStr[88][5] = "25"
AStr[88][6] = "27"

# Question Number 89
AStr[89][0] = "What is the result:"
```

```
     AStr[89][1] = "12 PLUS 12 = ?"
     AStr[89][2] = "24"
     AStr[89][3] = "22"
     AStr[89][4] = "20"
     AStr[89][5] = "26"
     AStr[89][6] = "28"

======
```

Conclusions and Future Work

During the coming months, the plan is to develop and implement the JavaScript and Python skeletons; and then continue with expanding the Quiz Games collection..

=====

Appendix A: New BDS Software License

This New Software License applies to all software found on the BDS Software site, and supersedes all previous copyright notices and licensing provisions which may appear in the software itself or in any documentation therefor.

All software which has previously been placed in the public domain remains in the public domain.

All other software, programs, experiments and reports, documentation, and any other material on this site (other than that attributed to outside sources) is hereby copyright © 2018 (or later if so marked) by M. David Johnson.

All software, documentation, and other information on the BDS Software site is available for you to freely download without cost.

Whether you downloaded such items directly from this site, or you obtained them by any other means, you are hereby licensed to copy them, to sell or give away such copies, to use them, and to excerpt from them, in any way whatsoever, so long as nothing you do with them would denigrate the name of our Lord and Savior, Jesus Christ.

I make absolutely no warranty whatsoever for any of these items. You use them entirely at your own risk.

If they don't work for you, I commiserate.

If they crash your system, I sympathize.

But I accept no responsibility whatsoever for any such consequences. Under no circumstances will BDS Software or M. David Johnson be liable for any negative results of any kind which you may experience from downloading or using these items.

BDS Software's former mail address at P.O. Box 485 in Glenview, IL is no longer valid. Any mail sent to that address will be rejected by the U.S. Postal Service. See my Contact page.

M.D.J. 2018/06/08

=====

Works Cited

PC Magazine. https://www.pcmag.com/encyclopedia/term/cross-platform

=====