CPSC3121 Program #4 See Calendar

# Check Digit – (Packed Arithmetic)

100 Points

Data Formats: In: CSU.PUBLIC.DATA(DATAPRG&PROG)

| | | | |
|---|---|---|---|
| CC | 1 - 8 | Variable | (Input and Output) |
| CC | 9 - 10 | Spaces | |
| CC | 11 - 13 | OK | (If check digit is correct) |
| | OR | | |
| CC | 11 - 13 | BAD | (If check digit was bad – show correct one!) |

Validate the input to insure it is 8 numeric digits! Either skip bad input or print it!
Check digit calculation: Multiply (odd) digits [1, 3, 5, 7] by 3 and add to even digits [2, 4, 6] giving sum. Subtract sum from next multiple of 10 not less than sum.

(See other side of page for several examples!)

Input: 00000291
Digits 1, 3, 5, and 7 times 3 = 27
Digits 2, 4, and 6 total = 2
Total 27 + 2 = 29
Check digit = 30 – 29 = 1

Input: 00001037
Digits 1, 3, 5, 7 times 3 = 12
Digits 2, 4, and 6 total = 0
Total 12 + 0 = 12
Check digit = 20 – 12 = 8

Name your program: PROG4 and leave it in your Library.
Break the program logic into about one page subroutines called with BAS.
Use: CSU.PUBLIC.DATA(DATAPRG&PROG) as the input dataset.
It's OK to leave 'debug' items in the print line!

---

| | |
|---|---|
| 00000291 OK | (Original input value with OK check digit) |
| 00000291 | (Copy of original value – not changed) |
| 00001037 BAD | (Original input value with BAD check digit) |
| 00001038 | (Input value with corrected check digit) |

# CPSC3121 Program #4 See Calendar

Another way to look at the check digit calculation is by creating a list of multiply factors.

Number: 0 0 0 0 1 2 6 C
Factors: 3 1 3 1 3 1 3 1
Products: 0 0 0 0 3 2 18
Sum: 3 + 2 + 18 = 23
Next multiple of 10 = 30
Check digit = 30 – 23 = 7

Number: 0 0 0 0 0 9 9 C
Factors: 3 1 3 1 3 1 3 1
Products: 0 0 0 0 0 9 27
Sum: 9 + 27 = 36
Next multiple of 10 = 40
Check digit = 40 – 36 = 4

Number: 0 0 0 0 0 2 9 C
Factors: 3 1 3 1 3 1 3 1
Products: 0 0 0 0 0 2 27
Sum: 2 + 27 = 29
Next multiple of 10 = 30
Check digit = 30 – 29 = 1

A much more general version of a check digit routine could easily be made by entering the factors on the first input record, or having a table of the most commonly used factors built into the program. A variable in the input record could select the appropriate set of factors. Some common factors for an eight digit number are:

2 1 3 2 1 3 2 1
8 7 6 5 4 3 2 1
2 1 2 1 2 1 2 1
3 1 3 1 3 1 3 1 (Bar code validation)