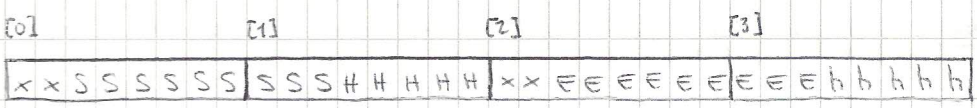
(ITEM 2 + ITEM Ø)

2012-09-03

INPUT

- RECORD DB 4 DUP (?)



x = unused

S = starting day [1-366] (9 bits) E = ending day [1-366] (9 bits)

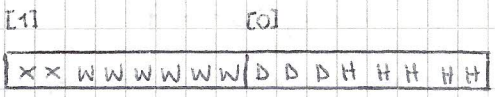
H = starting hour [0-23] (5 bits) h = ending hour [0-23] (5 bits)

- RATES DB 3 DUP (?) ← cost units on 8 bits

RATES[0] = hourly rate, RATES[1] = daily rate, RATES[2] = weekly rate

OUTPUT

- DURATION_OF_RENTAL DW ?



x = unused

W = #weeks of rental (6 bits)

D = #days of rental (3 bits)

H = #hours of rental (5 bits)

- COST_TO_BE_CHARGED DW ? ← overall cost of the rental

First compute the duration of the rental in W-D-H, and then use this result to compute the overal cost.

To compute the duration:

#D = (ENDING DAY - STARTING DAY) - 1

↑ do not consider the last day, because you still don't know if it is a whole or only a partial day of rental

#H = ENDING HOUR - STARTING HOUR

if #H = 0 ⟹ #D = #D + 1 (rental started and ended at the same hour ⟹ 24 hours = 1 day.)

if #H > 0 ⟹ #D = #D + 1

if #H < 0 ⟹ #D = #D and #H = 24 + #H

(e.g. starting hour = 9, ending hour = 7

⟹ #H = -2 ⟹ no another complete day, but only #H = 24 + #H = 24 + (-2) = 22 hours)

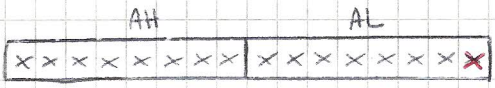
Compute TAX = 25% of COST_TO_BE_CHARGED

25% ≡ COST / 4 ≡ 2 SHR

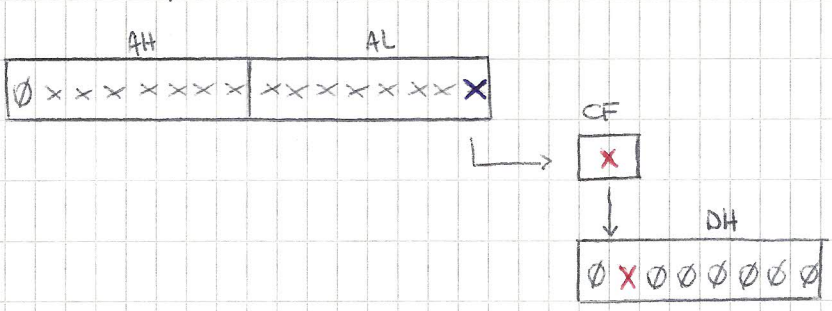
Since you want also to round the obtained value, you have also to store the fractional part, that is contained in the CF after a SHR.

AX ← COST_TO_BE_CHARGED

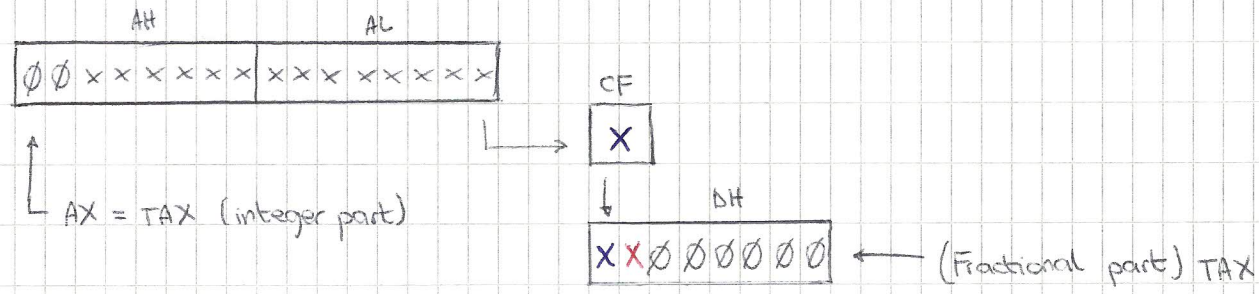
ROUNDING FRACTIONAL PART



SHR AX, 1



SHR AX, 1



0.5 = 10000000B (interpreted as fractional part)

IF DH ≥ 0.5 ⟹ AX = AX + 1

otherwise AX = AX + 0