

(ITEM 1D)

2014-09-15

N EQU 50

• EXPENSES DB 3*N DUP(?) ; at most 50 entries of 3 BYTES

entry:

[0]	[1]	[2]
V	D D D D D C C	E E E E E E E E F F F F F F F F

V=0 \Rightarrow NON VALID ENTRY, V=1 \Rightarrow VALID ENTRY

D = DAY OF MONTH (1-31)

C = CURRENCY 00 = USD, 01 = CAD, 10 = CHF, 11 = EUR

E = AMOUNT (integer part)

F = AMOUNT (fractional part)

• EXCHANGE 1B 32*3 DUP(?)

entry 0 \rightarrow AVERAGE exchange rateentry 1-31 \rightarrow exchange rate FOR EACH DAY (FORMAT: 0.TTTTTTTT)entry:

[0]	[1]	[2]
U U U U U U U U	A A A A A A A A	H H H H H H H H

 (fractional part)USD \rightarrow EUR CAD \rightarrow EUR CHF \rightarrow EUR

• CONVERTED DB 3+N DUP(?)

 \leftarrow same format of EXPENSES array

V = 1 (valid entry)

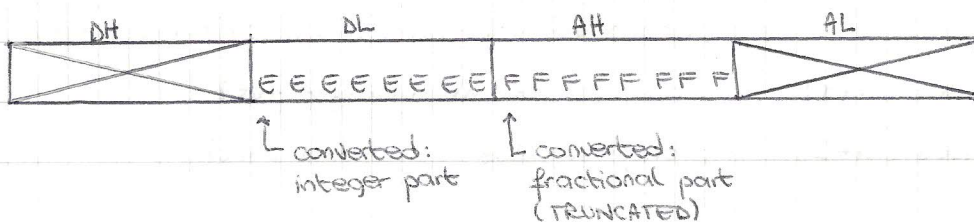
E.F \leftarrow VALUE CONVERTED IN EUR

CC = 11 (converted in eur)

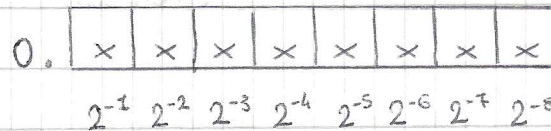
D = original day

CONVERSION:

(AX) 16 bits	<div style="display: flex; justify-content: space-between;">AHAL</div> <div style="text-align: center;">E E E E E E E E F F F F F F F F</div>	×	\leftarrow Original value
(CX) 16 bits	<div style="display: flex; justify-content: space-between;">CHCL</div> <div style="text-align: center;">0 0 0 0 0 0 0 0 T T T T T T T T</div>	=	\leftarrow Exchange rate

(DX:AX) 32 bits \leftarrow RESULT

FRACTIONAL PART ON 8 BITS



FOR A CORRECT PRINT OF THE FRACTIONAL PART, YOU MUST **DIVIDE IT BY 256** (8 SHR)

XOR SI, SI

convertloop:

MOV AL, EXPENSES[SI]

TEST AL, 10000000B ; check validity of the entry

JE endOfConversion

AND AL, 00000011B ; keep currency

CMP AL, 00000011B ; check if EUR

JE eurNoConversion

MOV AL, EXPENSES[SI]

AND AL, 01111100B ; keep date (index for EXCHANGE array)

MOV CONVERTED[SI], AL ; store the date

OR CONVERTED[SI], 10000011B ; set V=1 and CC=EUR

SHR AL, 1

SHR AL, 1 ; AL ← 00000000

MOV CL, 3 ; because each entry of EXCHANGE is on 3 BYTES

MUL CX ; <CAX> ← base index of EXCHANGE array

XOR DI, DI

ADD DI, AX

MOV AL, EXPENSES[SI]

AND AL, 00000011B ; select the correct EXCHANGE RATE

ADD DI, AX

MOV CL, EXCHANGE[DI] ; exchange rate

XOR CH, CH ; <CX> ← 00000000TTTTTTTT

MOV AH, EXPENSES[SI+1] ; original amount (integer part)

MOV AL, EXPENSES[SI+2] ; original value (fractional part)

MUL CX

MOV CONVERTED [SI+1], AL ; converted (integer part)

MOV CONVERTED [SI+2], AH ; converted (fractional part)

JMP nextConversion

endNoConversion:

MOV AL, EXPENSES [SI]

MOV CONVERTED [SI], AL

MOV AL, EXPENSES [SI+1]

MOV CONVERTED [SI+1], AL

MOV AL, EXPENSES [SI+2]

MOV CONVERTED [SI+2], AL

nextConversion:

; print old value and new value

ADD SI, 3

JMP convertloop

endOfConversion:

.exit

Example arrays:

EXPENSES

DAY 2 CHF
1 0 0 0 1 0 1 0 | 0 0 0 1 0 0 0 0 | 0 0 1 0 1 1 1 0

16,1796875 → 8,08984375

DAY 3 EUR
1 0 0 0 1 1 1 1 | 0 0 0 0 1 0 0 1 | 0 0 1 1 0 0 1 0

9,1953125 → 9,1953125

DAY 5 USD
1 0 0 1 0 1 0 0 | 1 1 1 0 1 0 1 1 | 0 1 1 0 0 0 0 1

235,37890625 → 58,8447265625

DAY 6 CAD
1 0 0 1 1 0 0 1 | 1 0 0 0 0 0 0 0 | 0 0 1 0 0 0 0 0

128,125 → 96,09375

EXCHANGE

USD CAD CHF
[0] 0 0 0 0 0 1 0 1 | 0 0 0 0 1 0 1 0 | 0 0 0 0 0 0 0 1

[1] 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0

[2] 0 1 1 1 0 0 0 1 | 0 1 0 0 0 0 0 0 | 1 0 0 0 0 0 0 0
0.5

DAY 2 : CHF

[3] 0 0 0 0 1 0 0 0 | 0 0 0 0 0 0 0 1 | 1 0 0 0 0 0 0 0

DAY 3 : EUR (no conversion)

[4] 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0

[5] 0 1 0 0 0 0 0 0 | 1 0 0 0 0 0 0 0 | 1 1 1 0 0 0 0 0
0.25

DAY 5 : USD

[6] 1 0 1 0 0 0 0 0 | 1 1 0 0 0 0 0 0 | 1 0 0 0 0 0 0 0
0.75

DAY 6 : CAD