TASK 1:

[org 0x100]

jmp start

multiplicand: db 17

multiplier: db 6

result: db 0

start: mov cl, 4

mov bl, [multiplicand]

mov dl, [multiplier]

checkbit: shr dl, 1

jnc skip

add [result], bl

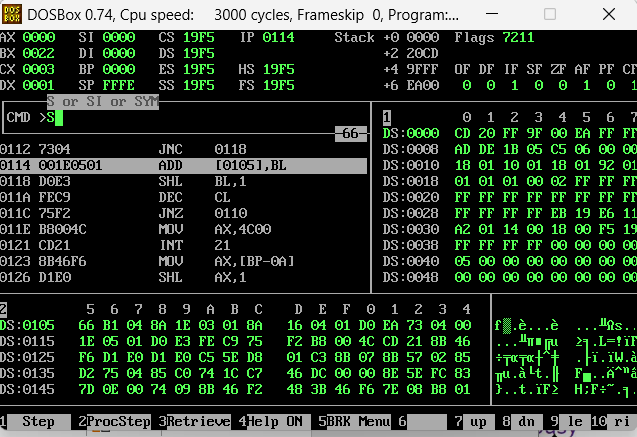
skip: shl bl, 1

dec cl

jnz checkbit

mov ax, 0x4c00

int 0x21



TASK 3:

[org 0x100]

jmp start

multiplicand: dd 1300

multiplier: dw 500

result: dd 0

start: mov cl, 16

mov dx, [multiplier]

checkbit: shr dx, 1

jnc skip

mov ax, [multiplicand]

add [result], ax

mov ax, [multiplicand+2]

adc [result+2], ax

skip: shl word [multiplicand], 1

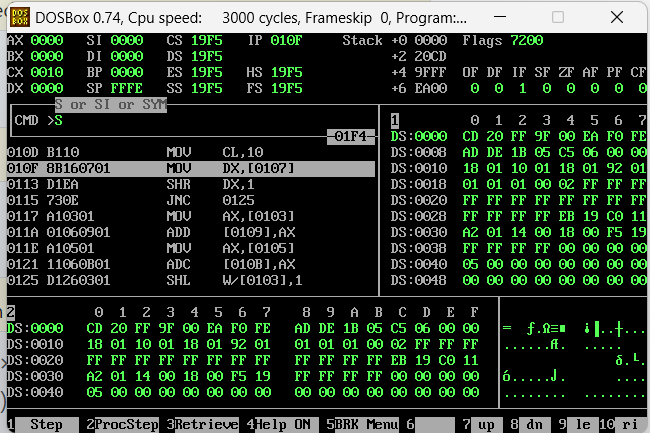
rcl word [multiplicand+2], 1

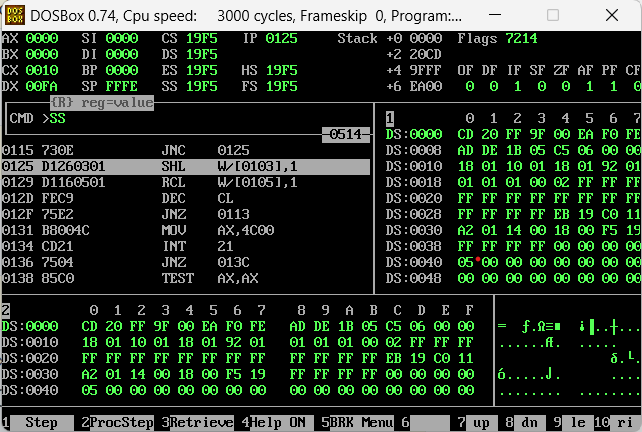
dec cl

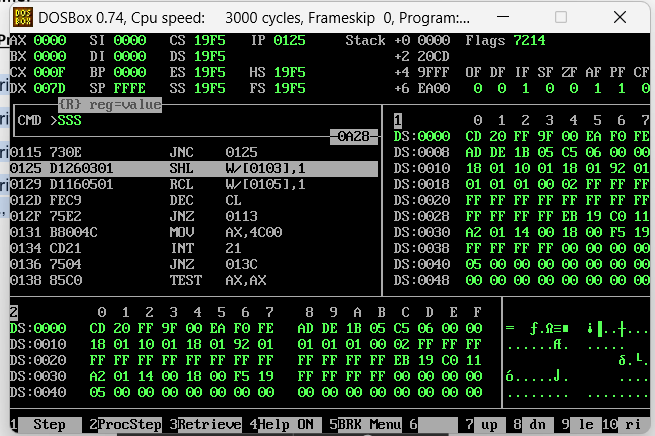
jnz checkbit

mov ax, 0x4c00

int 0x21







TASK 4:

**JA AND JG :**

JA means "jump if Carry Flag unset and Zero Flag unset".

The CMP instruction is really the same as the sub instruction (i.e. it subtracts its arguments), except that the result is not saved but only the condition flags are updated.

If we were comparing unsigned integers, subtracting (a-b) sets the Carry Flag if b is greater than a, and the Zero Flag if b is equal to a, so if neither of these flags is set, it follows that a is greater than b.

If we wanted a comparison of signed numbers, we'd have to compare the Sign Flag (i.e. the topmost bit of the result) to the Overflow Flag, and check that the Zero Flag is unset, which is what the JG instruction does.

Thus, the CMP instruction does not care about whether the arguments are signed or unsigned. This distinction is only in how the flags are interpreted afterwards.

After a CMP (Compare) of two unsigned (32 bits) integers, should I use JL (Jump if Less) or JB (Jump if Below) to see which one is the largest of the two.  
I know that the terms “above” and “below” are used for unsigned integers but since CMP does a subtraction, the (internal) result can be negative and therefore signed .. This has me confused because how does CMP 'know' that a certain 32 bits memory address is supposed to hold signed or unsigned integers ?  
we use JL, JG when comparing signed integers,  
we use JA, JB when comparing unsigned integers,  
because CMP actually does 2 compares: one for signed and one for unsigned integers.

**JA JUMP EXAMPLE**

cmp eax, edx

ja somewhere ; will go to somewhere if eax >u edx

; where >u is "unsigned greater than"

**JG JUMP EXAMPLE**

cmp eax, edx

jg somewhere ; will go to somewhere if eax >s edx

; where >s is "signed greater than"