TASK 1:

; bubble sort subroutine called twice

[org 0x0100]

jmp start

data: dw 60, 55, 45, 50, 40, 35, 25, 30, 10, 0

data2: dw 328, 329, 898, 8923, 8293, 2345, 10, 877, 355, 98

dw 888, 533, 2000, 1020, 30, 200, 761, 167, 90, 5

swap: db 0

bubblesort: dec cx

shl cx, 1

mainloop: mov si, 0

mov byte [swap], 0

innerloop: mov ax, [bx+si]

cmp ax, [bx+si+2]

jbe noswap

mov dx, [bx+si+2]

mov [bx+si], dx

mov [bx+si+2], ax

mov byte [swap], 1

noswap: add si, 2

cmp si, cx

jne innerloop

cmp byte [swap], 1

je mainloop

ret

start: mov bx, data

mov cx, 10

call bubblesort

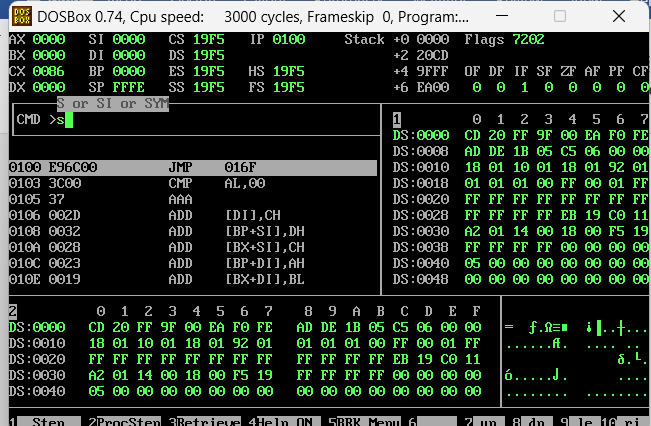
mov bx, data2

mov cx, 20

call bubblesort

mov ax, 0x4c00

int 0x21



A screenshot of a computer

Description automatically generated with medium confidence

TASK 2:

; bubble sort subroutine taking parameters from stack

[org 0x0100]

jmp start

data: dw 88, 47, 0, -5, 83, 10, 29, -30, 2, 64

data2: dw 328, 329, 898, 8923, 8293, 2345, 10, 877, 355, 98

dw 1024, 88, 351, 743, 994, 125, 753, 852, 1, 81

swapflag: db 0

bubblesort: push bp

mov bp, sp

push ax

push bx

push cx

push si

mov bx, [bp+6]

mov cx, [bp+4]

dec cx

shl cx, 1

mainloop: mov si, 0

mov byte [swapflag], 0

innerloop: mov ax, [bx+si]

cmp ax, [bx+si+2]

jnle noswap

xchg ax, [bx+si+2]

mov [bx+si], ax

mov byte [swapflag], 1

noswap: add si, 2

cmp si, cx

jne innerloop

cmp byte [swapflag], 1

je mainloop

pop si

pop cx

pop bx

pop ax

pop bp

ret 4

start: mov ax, data

push ax

mov ax, 10

push ax

call bubblesort

mov ax, data2

push ax

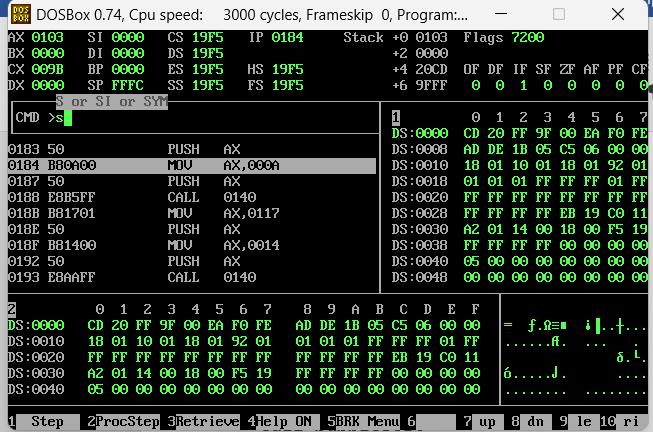
mov ax, 20

push ax

call bubblesort

mov ax, 0x4c00 ; terminate program

int 0x21



TASK 3:

[org 0x100]

jmp start

multiplicand: dd 1024

multiplier: dw 381

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

