## Second Largest:

[org 0x0100]

jmp start

data: dw 1,5,6,8,3,4,2,3

length: dw 8

max: dw 0

secMAx: dw 0

if:

mov [secMAx] , ax

mov ax , [data + bx]

mov [max] , ax

add bx , 2

sub cx , 1

jz terminate

jnz compareloop

start:

mov ax , [data + bx]

mov [max] , ax

mov bx , 2

mov cx , [length]

compareloop:

cmp ax , [data + bx]

jnae if

else:

cmp [secMAx] , ax

ja l1

add bx , 2

sub cx , 1

jz terminate

jnz compareloop

l1:

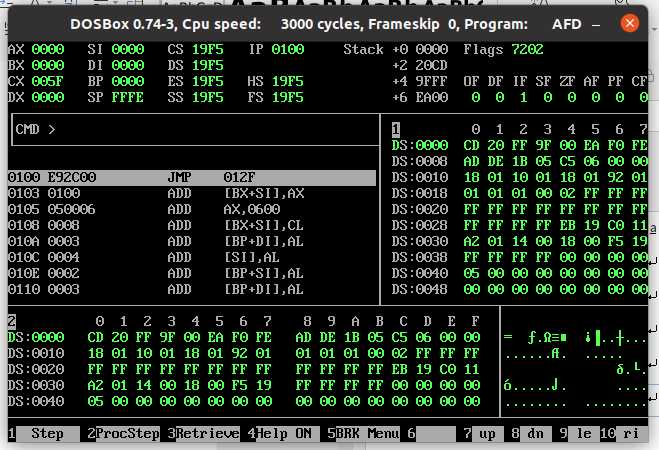
mov [secMAx] , ax

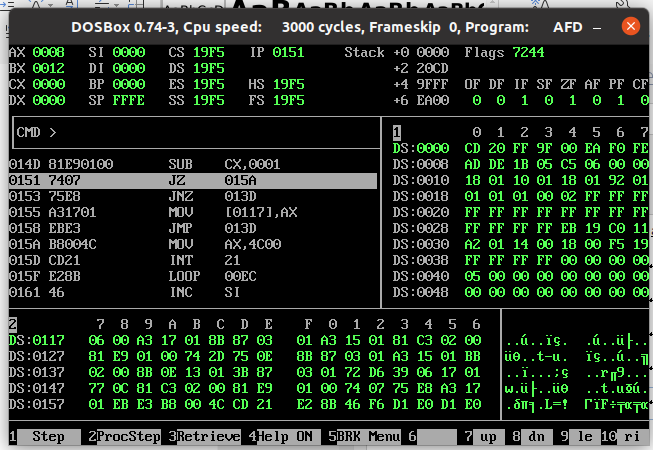
jmp compareloop

terminate:

mov ax , 0x4c00

int 0x21





## Second Smallest:

[org 0x0100]

jmp start

data: dw 3,1,6,2,1,7,4,9

length: dw 8

min: dw 0

secMin: dw 0

if:

mov [secMin] , ax

mov ax , [data + bx]

mov [min] , ax

add bx , 2

sub cx , 1

jz terminate

jnz compareloop

start:

mov ax , [data + bx]

mov [min] , ax

mov bx , 2

mov cx , [length]

compareloop:

cmp ax , [data + bx]

jnbe if

else:

cmp [secMin] , ax

jb l1

add bx , 2

sub cx , 1

jz terminate

jnz compareloop

l1:

mov [secMin] , ax

jmp compareloop

terminate:

mov ax , 0x4c00

int 0x21

