COMP LAB 1

ADIT LUHADIA

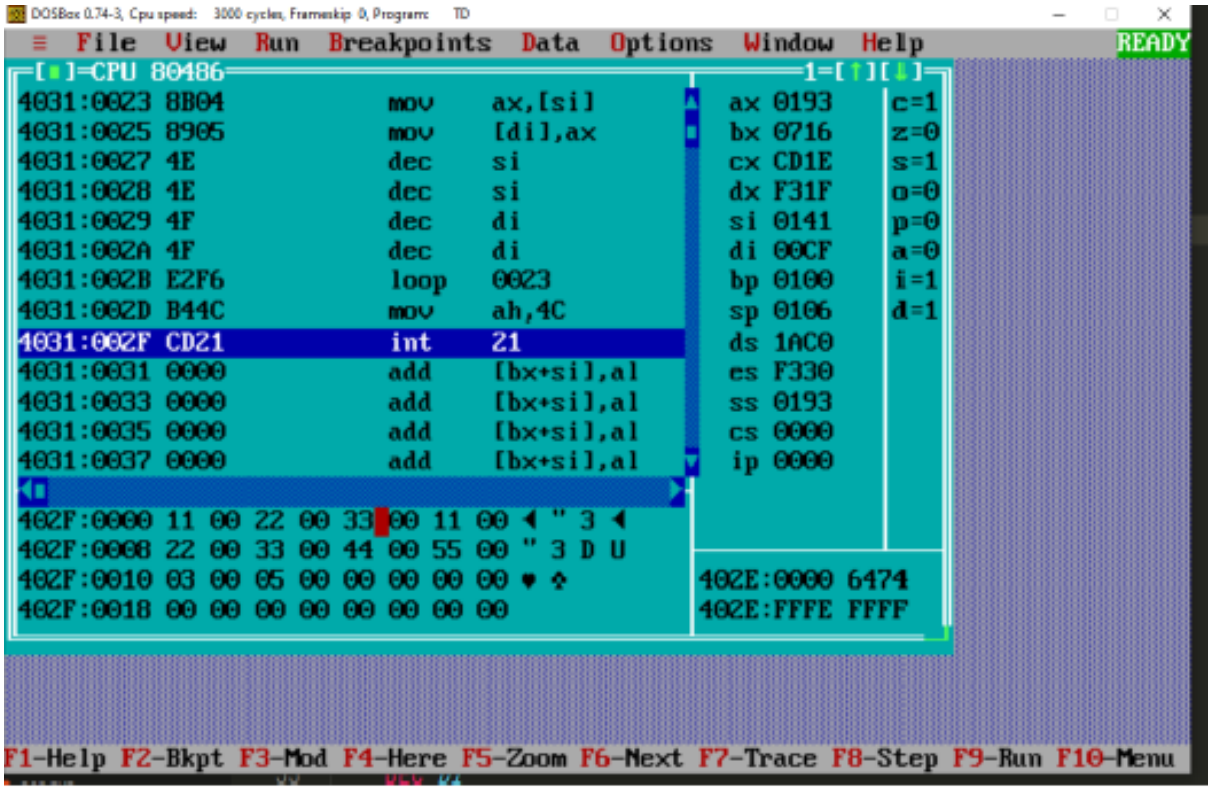
190911112

IT A

Q1 A) Transfer 5 words of data in data segment, where blocks are a). Overlapping and b).non-overlapping.

```
DATA
SEGMENT
        SRC DW 11H, 22H, 33H, 44H, 55H
        DES DW 3 DUP (0)
        DES_START DW 3
        COUNT DB 5
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV CH, 00H
    MOV CL, COUNT

    LEA SI, SRC
    ADD SI, CX
    ADD SI, CX
    DEC SI
    DEC SI
    LEA DI, DES
    ADD DI, DES_START
    ADD DI, DES_START
    DEC DI
    DEC DI

COPY:
    MOV AX, [SI]
    MOV [DI], AX
    DEC SI
    DEC SI
    DEC DI
    DEC DI
    LOOP COPY
```

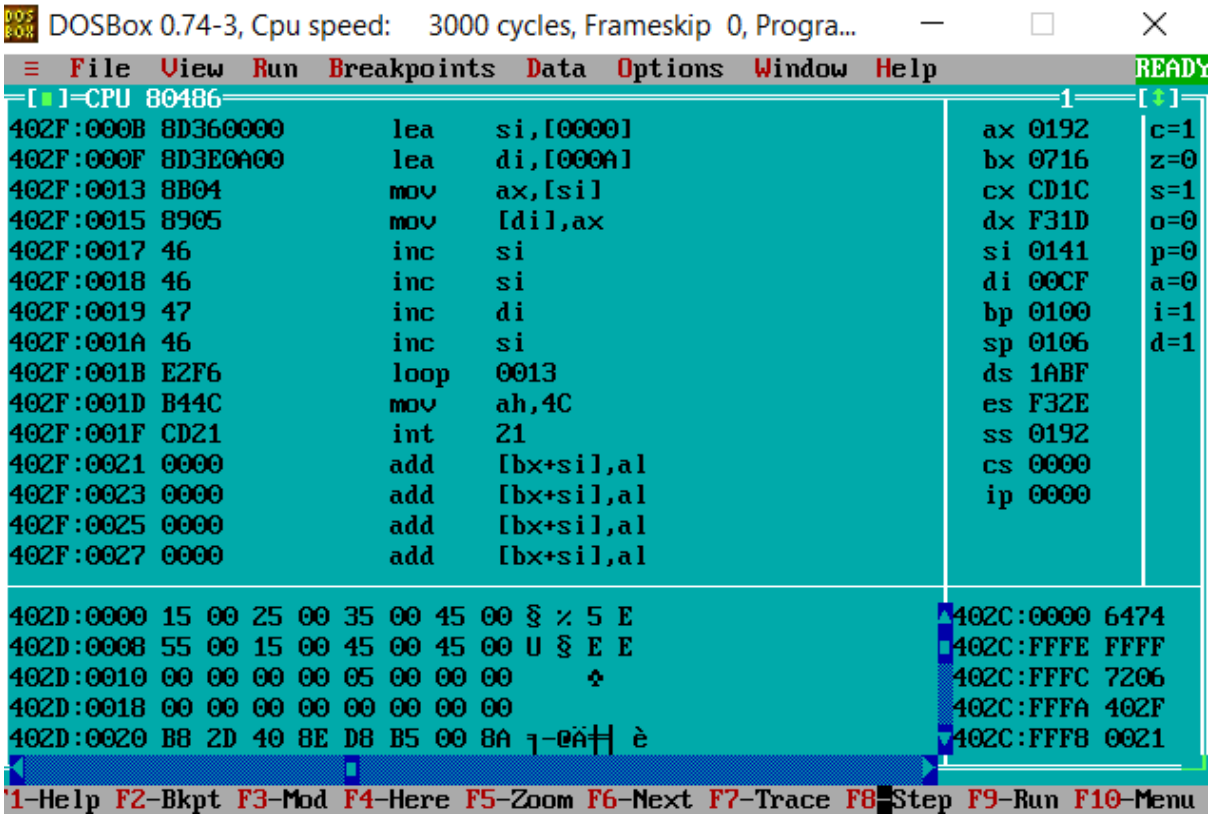
```
    MOV AH, 4CH
    INT 21H

CODE ENDS
END START
```



Q1 B)

```
DATA
SEGMENT
    SRC DW 15H, 25H, 35H, 45H, 55H
    DES DW 5 DUP(0)
    COUNT DB 5
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV CH, 00H
    MOV CL, COUNT
```

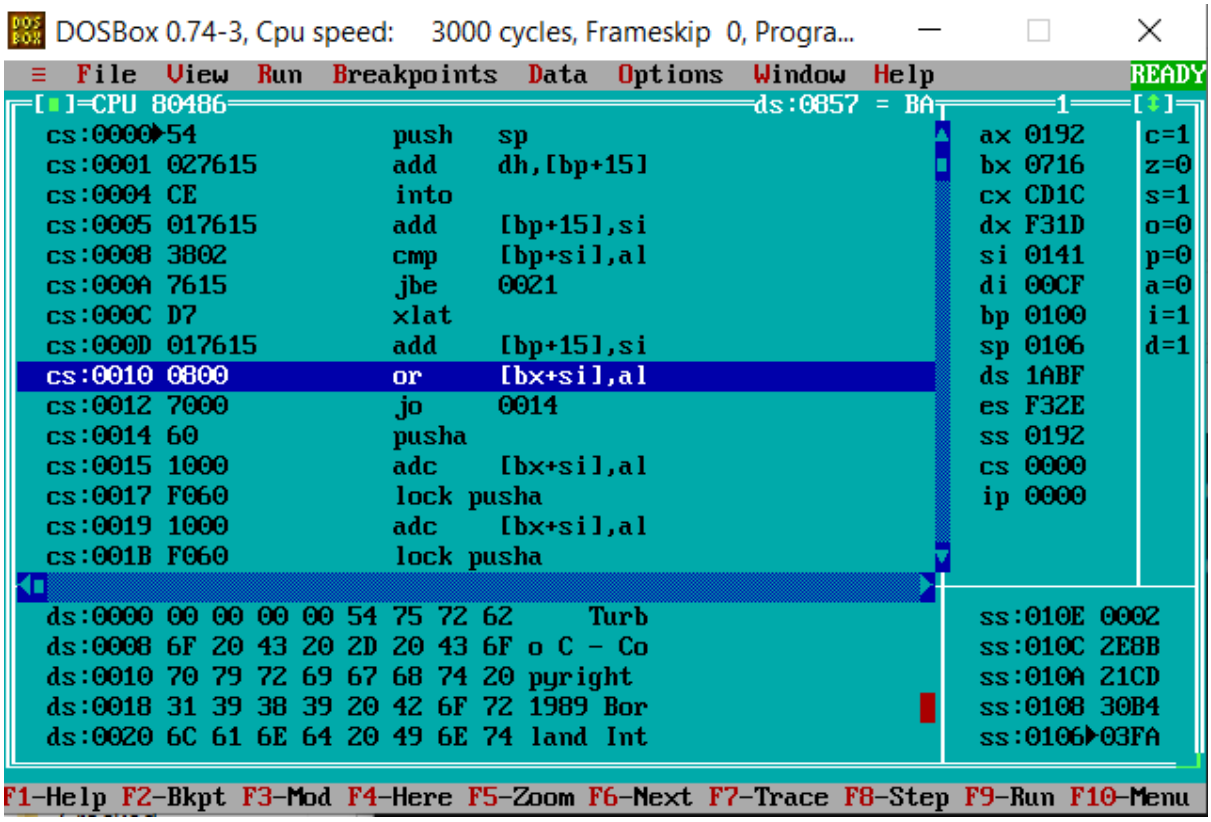
```
    LEA SI, SRC
    LEA DI, DES

COPY:
    MOV AX, [SI]
    MOV [DI], AX
    INC SI
    INC SI
    INC DI
    INC DI
    LOOP COPY

    MOV AH, 4CH
    INT 21H

CODE ENDS
END START
```



Q2) Add two 32 - bit hexadecimal numbers in data segment and store the result in data segment.

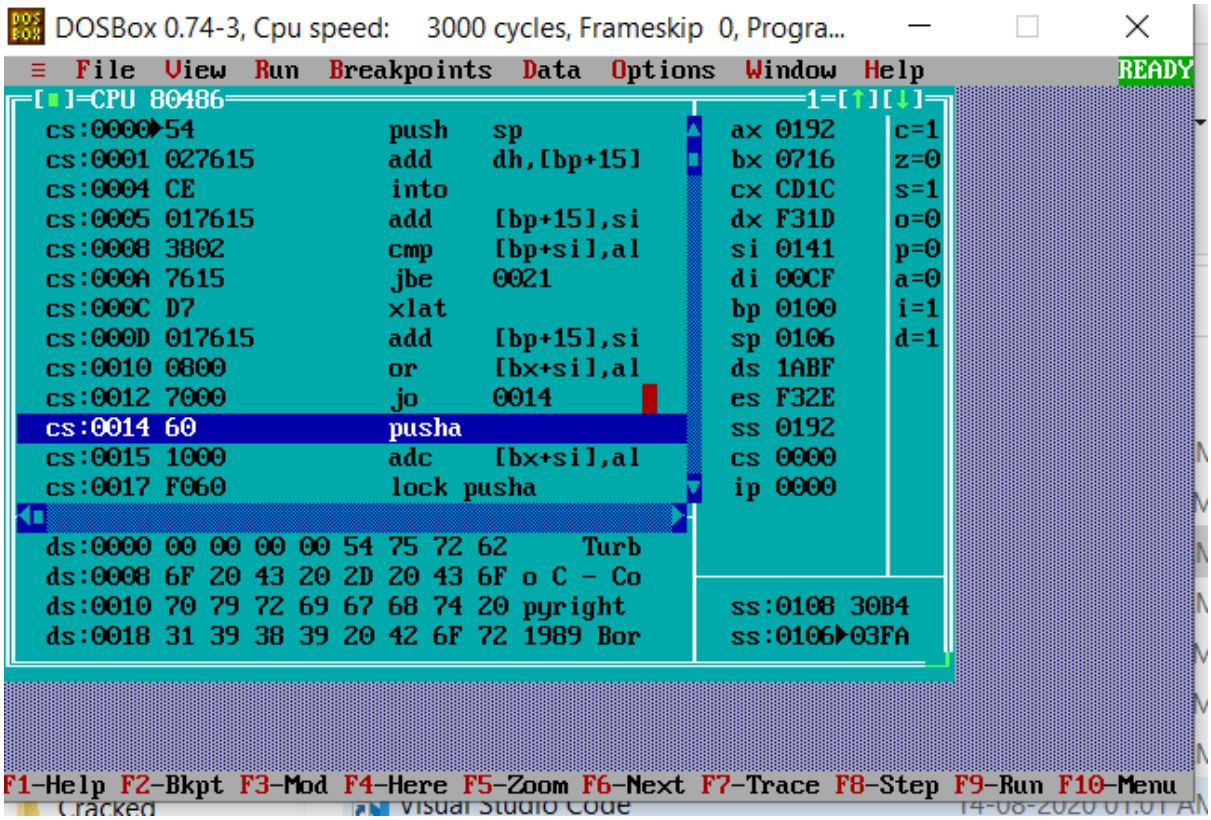
```
DATA
SEGMENT
    X DD 22099720H
    Y DD 88745474H
```

```
    RES DD ?
    CARRY DB ?
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV AX, WORD PTR X
    MOV BX, WORD PTR Y
    ADD AX, BX
    MOV WORD PTR RES, AX
    MOV AX, WORD PTR X+2
    MOV BX, WORD PTR Y+2
    ADC AX, BX
    MOV WORD PTR RES+2, AX
    ADC CARRY, 0

    MOV AH, 4CH
    INT 21H
CODE ENDS
END START
```



Q3) Subtract two 32 - bit numbers in data segment and store the result in data segment

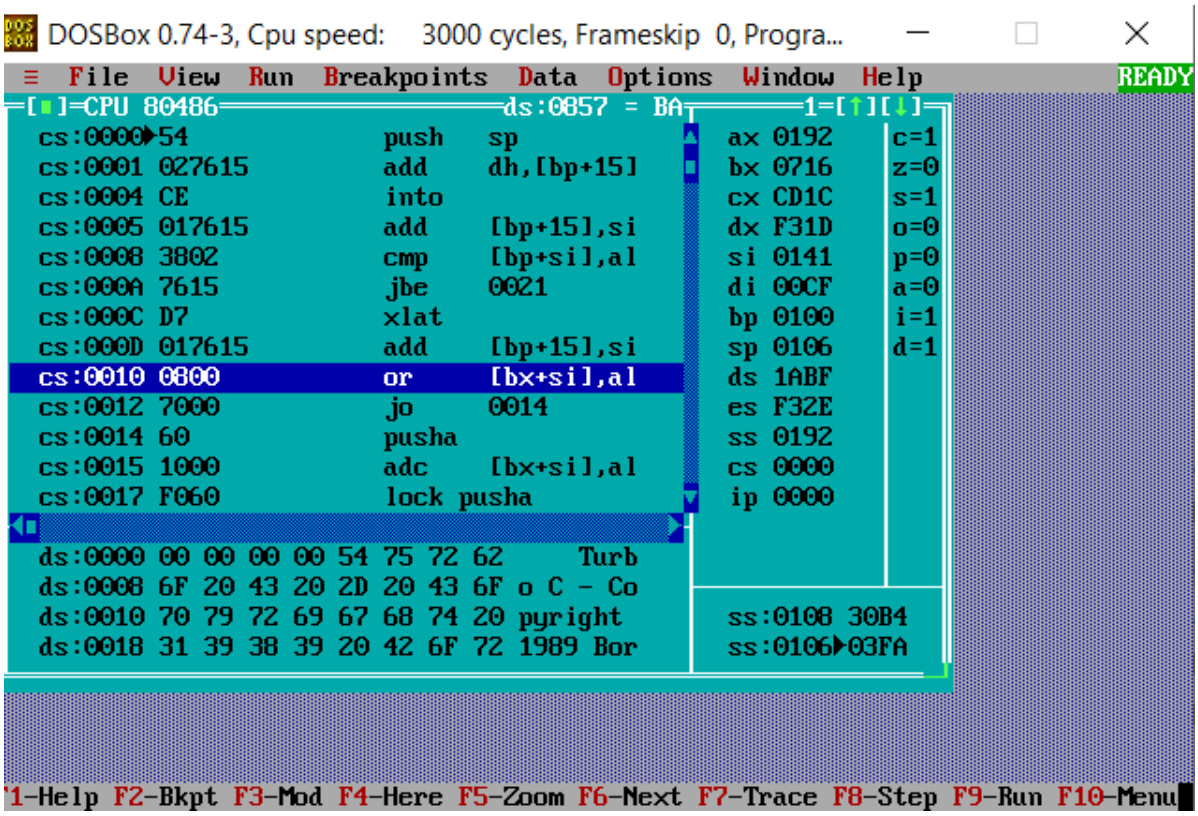
```
DATA
SEGMENT
        X DD 15489657H
        Y DD 12487596H
        RES DD ?
        CARRY DB ?
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV AX, WORD PTR X
    MOV BX, WORD PTR Y
    SUB AX, BX
    MOV WORD PTR RES, AX
    MOV AX, WORD PTR X+2
    MOV BX, WORD PTR Y+2
    SBB AX, BX
    MOV WORD PTR RES+2, AX
    ADC CARRY, 0

    MOV AH, 4CH
    INT 21H
CODE ENDS
END START
```



Q4) Add two 64 – bit decimal numbers in data segment and store the result in data segment

```
DATA
SEGMENT
        X DQ 9876543298765432H
        Y DQ 1234567812345678H
        RES DQ ?
        CARRY DB ?
    DATA ENDS

    CODE SEGMENT
        ASSUME CS: CODE, DS: DATA
    START:
        MOV AX, DATA
        MOV DS, AX

        MOV AX, WORD PTR X
        MOV BX, WORD PTR Y
        ADD AX, BX
        MOV WORD PTR RES, AX

        MOV AX, WORD PTR X+2
        MOV BX, WORD PTR Y+2
        ADC AX, BX
```

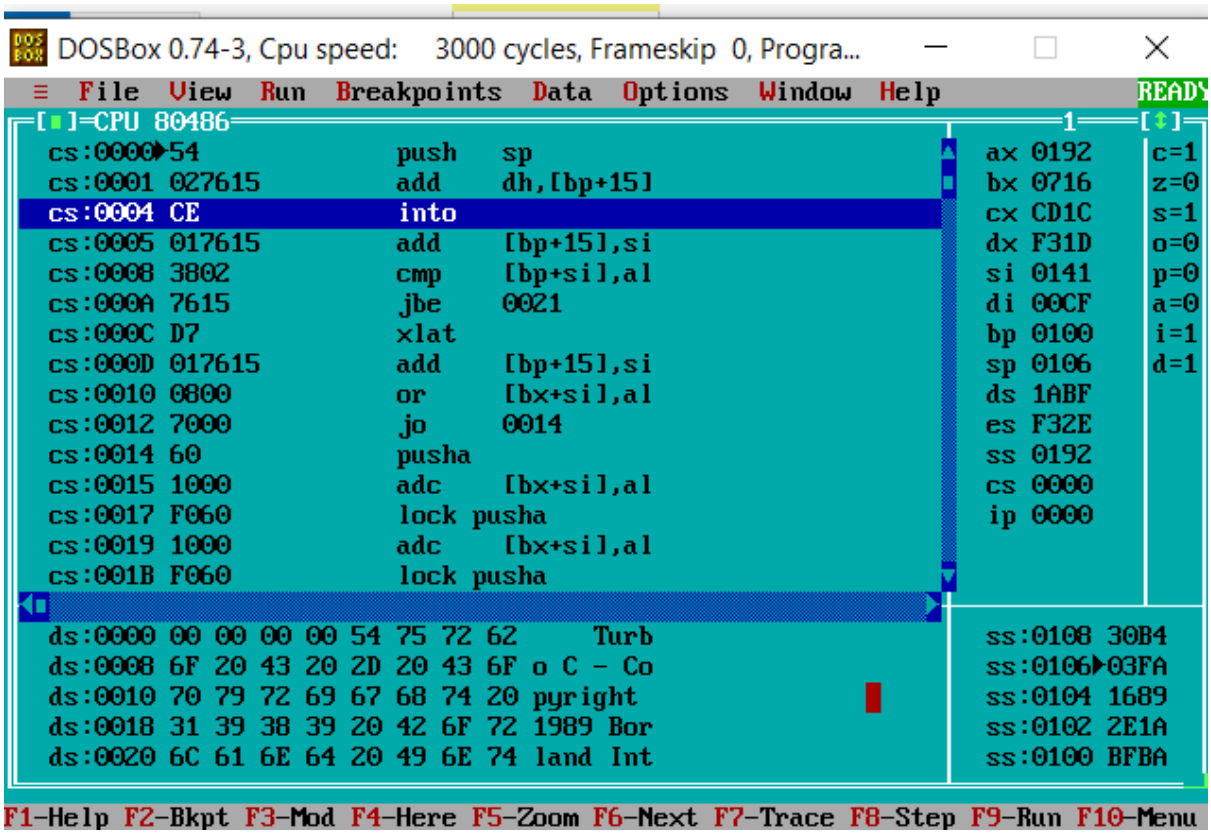
```
        MOV WORD PTR RES+2, AX

        MOV AX, WORD PTR X+4
        MOV BX, WORD PTR Y+4
        ADC AX, BX
        MOV WORD PTR RES+4, AX

        MOV AX, WORD PTR X+6
        MOV BX, WORD PTR Y+6
        ADC AX, BX
        MOV WORD PTR RES+6, AX

        ADC CARRY, 0

        MOV AH, 4CH
        INT 21H
    CODE ENDS
    END START
```



Q5) Find the sum of 10 unsigned bytes in an array and store the result in data segment

```
DATA
SEGMENT
        SRC DB 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH
        COUNT DB 10
```

```
    RES DW ?
DATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV CH, 00
    MOV CL, COUNT

    LEA SI, SRC
    MOV BX, 0

    CLC
SUM:
    mov AX, 0
    MOV AL, BYTE PTR [SI]
    ADC BX, AX
    INC SI
    LOOP SUM
    MOV RES, BX

    MOV AH, 4CH
    INT 21H

CODE ENDS
END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 54 push sp
cs:0001 027615 add dh,[bp+15]
cs:0004 CE into
cs:0005 017615 add [bp+15],si
cs:0008 3802 cmp [bp+si],al
cs:000A 7615 jbe 0021
cs:000C D7 xlat
cs:000D 017615 add [bp+15],si
cs:0010 0800 or [bx+si],al
cs:0012 7000 jo 0014
cs:0014 60 pusha
cs:0015 1000 adc [bx+si],al
cs:0017 F060 lock pusha
cs:0019 1000 adc [bx+si],al
cs:001B F060 lock pusha
ax 0192
bx 0716
cx CD1C
dx F31D
si 0141
di 00CF
bp 0100
sp 0106
ds 1ABF
es F32E
ss 0192
cs 0000
ip 0000
c=1
z=0
s=1
o=0
p=0
a=0
i=1
d=1
ds:0000 00 00 00 00 54 75 72 62 Turb
ds:0008 6F 20 43 20 2D 20 43 6F o C - Co
ds:0010 70 79 72 69 67 68 74 20 pyright
ds:0018 31 39 38 39 20 42 6F 72 1989 Bor
ds:0020 6C 61 6E 64 20 49 6E 74 land Int
ss:0108 30B4
ss:0106 03FA
ss:0104 1689
ss:0102 2E1A
ss:0100 BFBA
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu