LAB#06

LOOP

Objective

To understand the concepts of loop in assembly language

Theory

Loop

A loop is a sequence of instructions that is repeated. The number of times to repeat may be known in advance, or it may depend on conditions i.e. it• s a count controlled loop.

FOR Loop

This is a loop structure in which the loop statements are repeated a known number of times.

Keyword: LOOP

A FOR loop is implemented using the LOOP instruction. The counter for the loop is the CX register, which is initialized to loop count, which is the number of times the loop is executed. Execution of the LOOP instruction causes CX to be decremented automatically. If CX becomes 0,the next instruction after loop is done.

Sample Program:

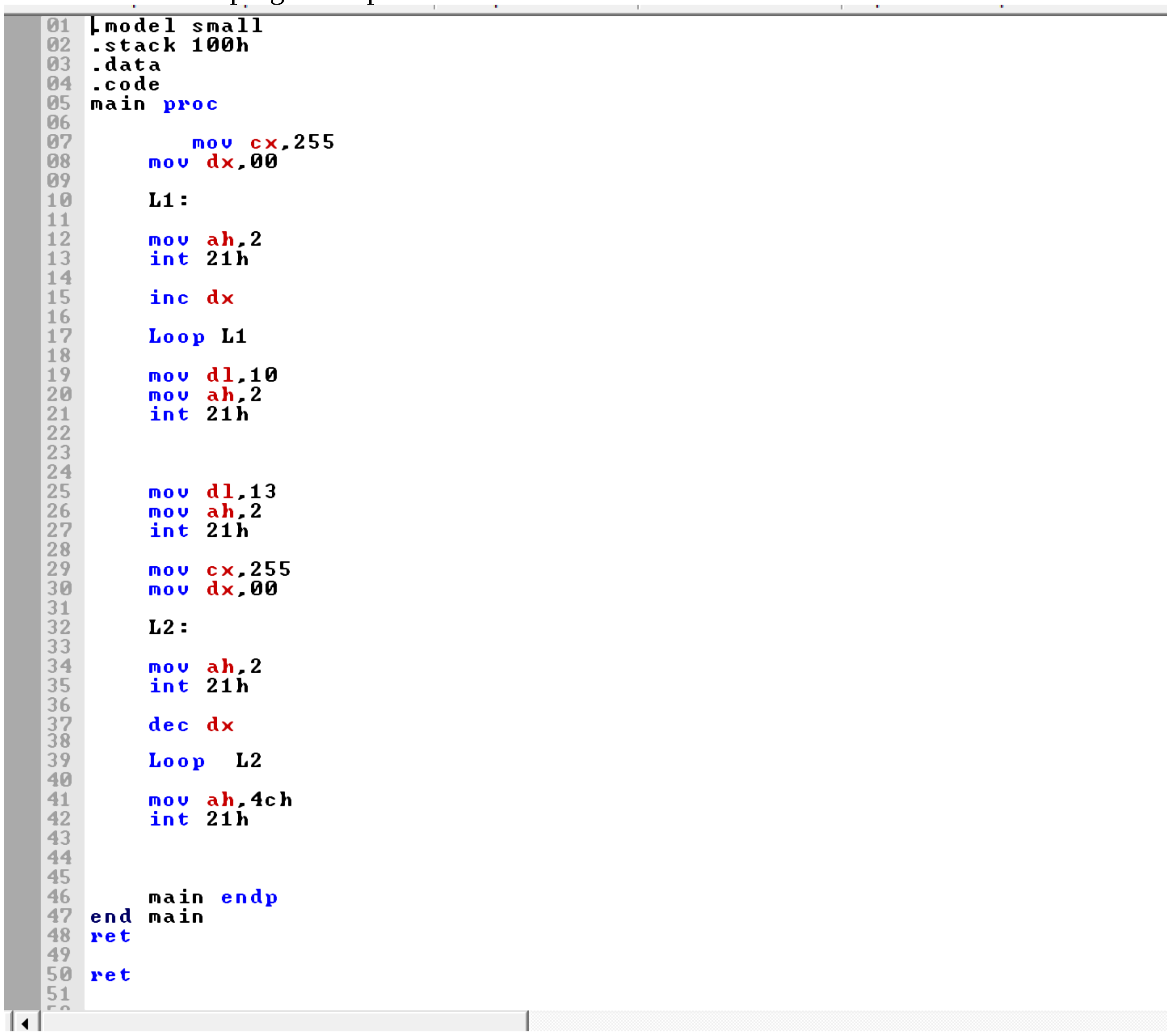
SOURCE CODE:

Object: Title a program that prints a character 100 times.

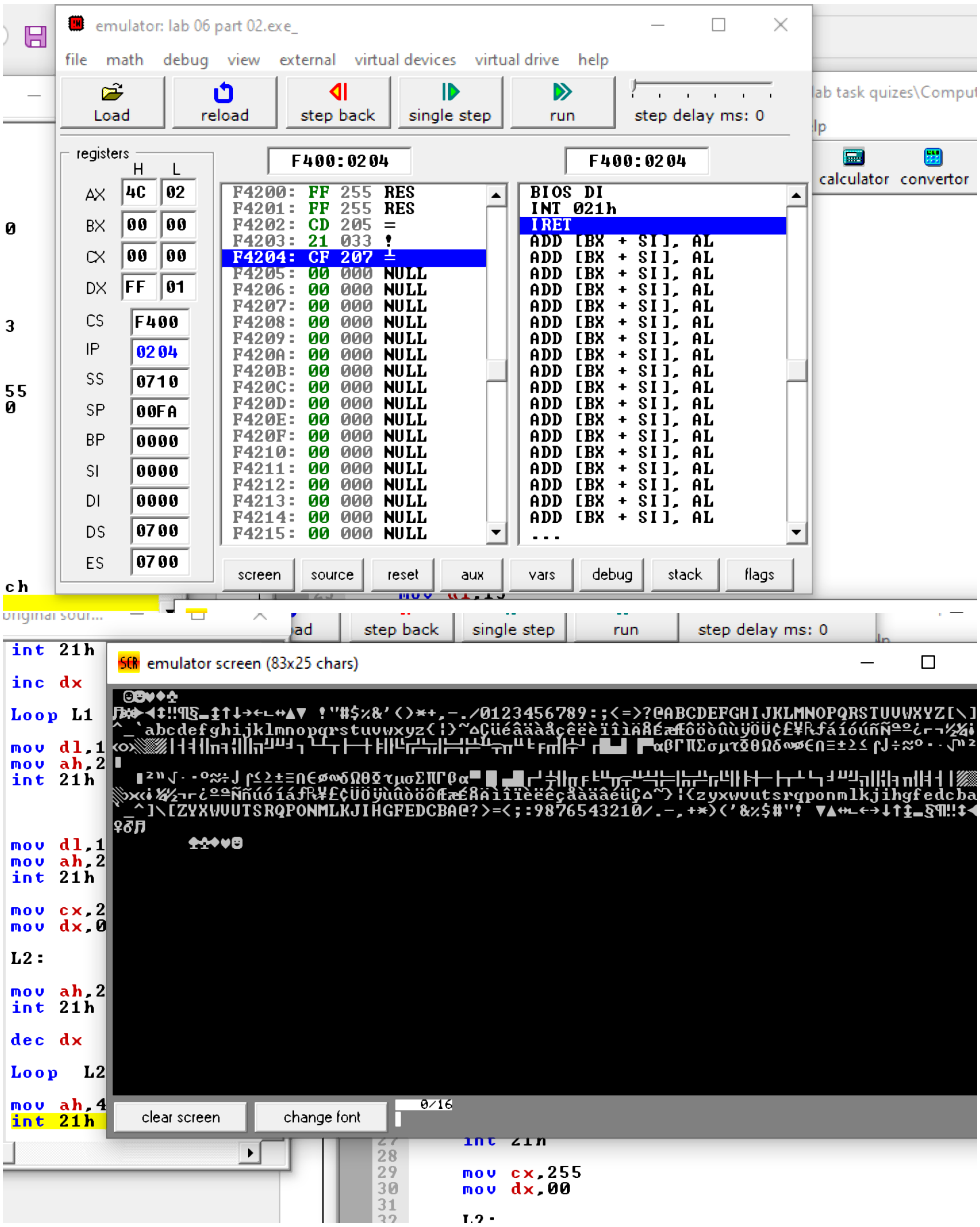
```
.model small
.stack 100h
.data
.code

mov ah, 02h ;display a character
mov cx, 100 ;number of times loop will execute
mov dl, „*•  ;ASCII code of character 0
print: ;loop starts from here
int 21h
loop print ;executes the FOR loop

.exit
endL
```

Lab Task:

Task#01: Write a program to print ASCII characters**.**



```
.model small
.stack 100h
.data
.code
main proc

        mov cx,255
    mov dx,00

    L1:

    mov ah,2
    int 21h

    inc dx

    Loop L1

    mov dl,10
    mov ah,2
    int 21h



    mov dl,13
    mov ah,2
    int 21h

    mov cx,255
    mov dx,00

    L2:

    mov ah,2
    int 21h

    dec dx

    Loop  L2

    mov ah,4ch
    int 21h



    main endp
end main
ret

ret
```


emulator: lab 06 part 02.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
calculator convertor
screen source reset aux vars debug stack flags
emulator screen (83x25 chars)
clear screen change font
0/16

registers

	H	L
AX	4C	02
BX	00	00
CX	00	00
DX	FF	01
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
```

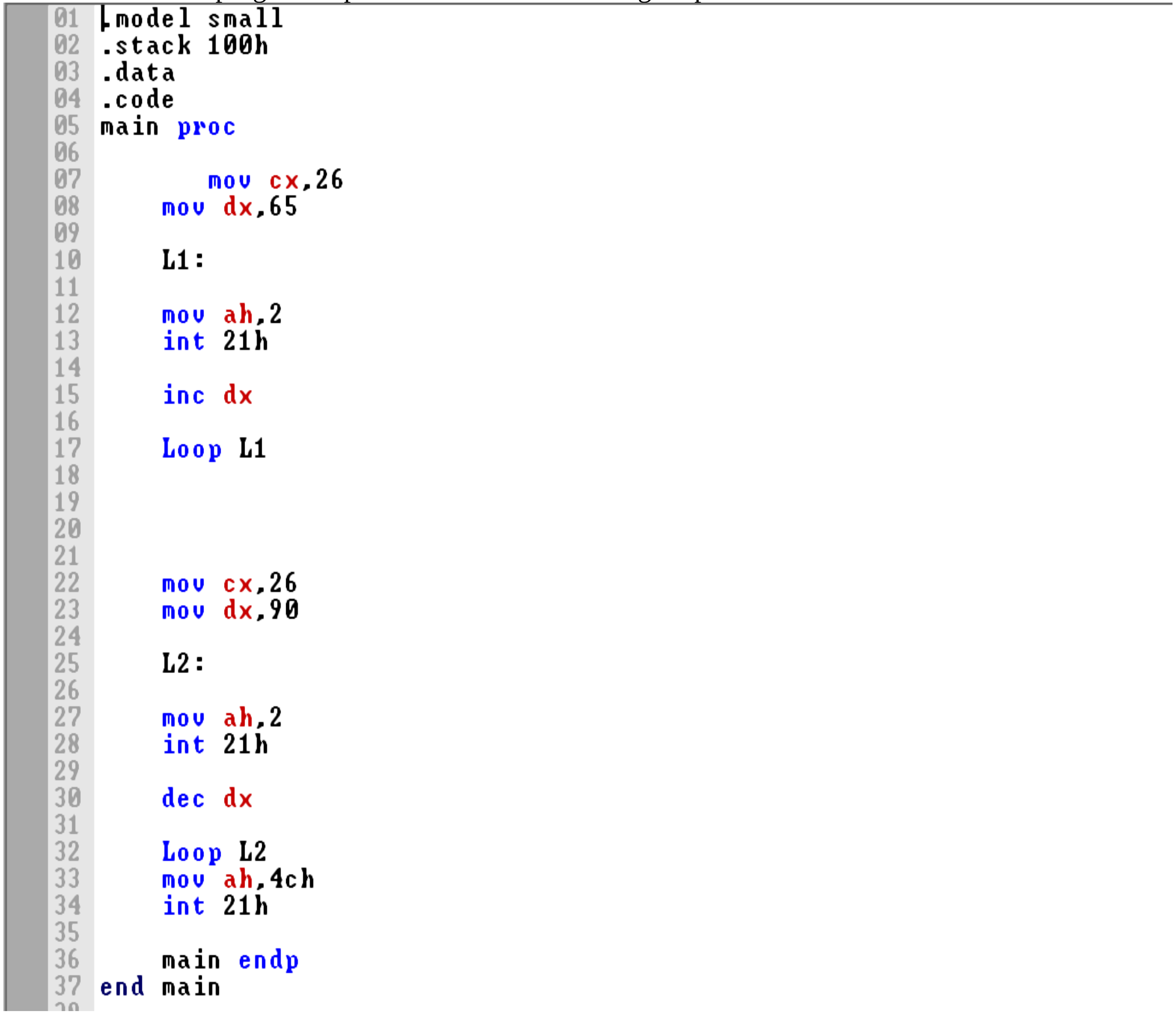
```
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

```
0

3

55
0

ch
int 21h

inc dx

Loop L1

mov dl,1
mov ah,2
int 21h


mov dl,1
mov ah,2
int 21h

mov cx,2
mov dx,0

L2:

mov ah,2
int 21h

dec dx

Loop L2

mov ah,4
int 21h
```

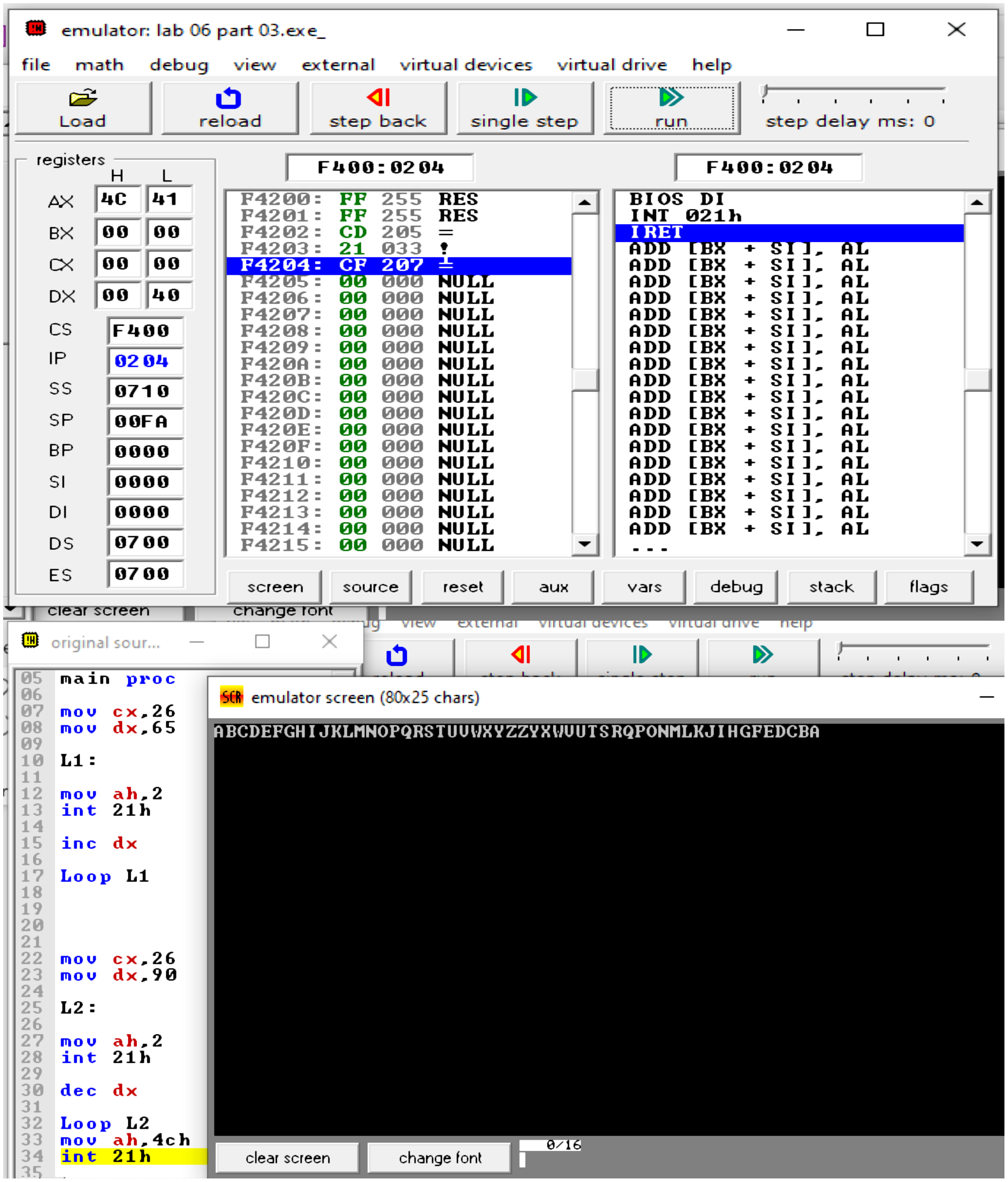
```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~⌂ÇüéâäàåçêëèïîìÄÅÉæÆôöòûùÿÖÜ¢£¥₧ƒáíóúñÑªº¿⌐¬½¼
αßΓπΣσµτΦΘΩδ∞φε∩≡±≥≤⌠⌡÷≈°∙·√ⁿ²
▐²ⁿ√·∙°≈÷⌡⌠≤≥±≡∩εφ∞δΩΘΦτµσΣπΓßα
¼½¬⌐¿ºªÑñúóíáƒ₧¥£¢ÜÖÿùûòöôÆæÉÅÄìîïèëêçåàäâéüÇ⌂~}|{zyxwvutsrqponmlkjihgfedcba
`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)('&%$#"! 
```

```
mov cx,255
mov dx,00

L2:
```

Task#02: Write a program to print A to Z character using loop



```
.model small
.stack 100h
.data
.code
main proc

        mov cx,26
    mov dx,65

    L1:

    mov ah,2
    int 21h

    inc dx

    Loop L1



    mov cx,26
    mov dx,90

    L2:

    mov ah,2
    int 21h

    dec dx

    Loop L2
    mov ah,4ch
    int 21h

    main endp
end main
```


emulator: lab 06 part 03.exe_

registers: AX 4C 41, BX 00 00, CX 00 00, DX 00 40, CS F400, IP 0204, SS 0710, SP 00FA, BP 0000, SI 0000, DI 0000, DS 0700, ES 0700

```
F400:0204

F4200:  FF 255  RES
F4201:  FF 255  RES
F4202:  CD 205  =
F4203:  21 033  !
F4204:  CF 207  ╧
F4205:  00 000  NULL
F4206:  00 000  NULL
F4207:  00 000  NULL
F4208:  00 000  NULL
F4209:  00 000  NULL
F420A:  00 000  NULL
F420B:  00 000  NULL
F420C:  00 000  NULL
F420D:  00 000  NULL
F420E:  00 000  NULL
F420F:  00 000  NULL
F4210:  00 000  NULL
F4211:  00 000  NULL
F4212:  00 000  NULL
F4213:  00 000  NULL
F4214:  00 000  NULL
F4215:  00 000  NULL
```

```
F400:0204

BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

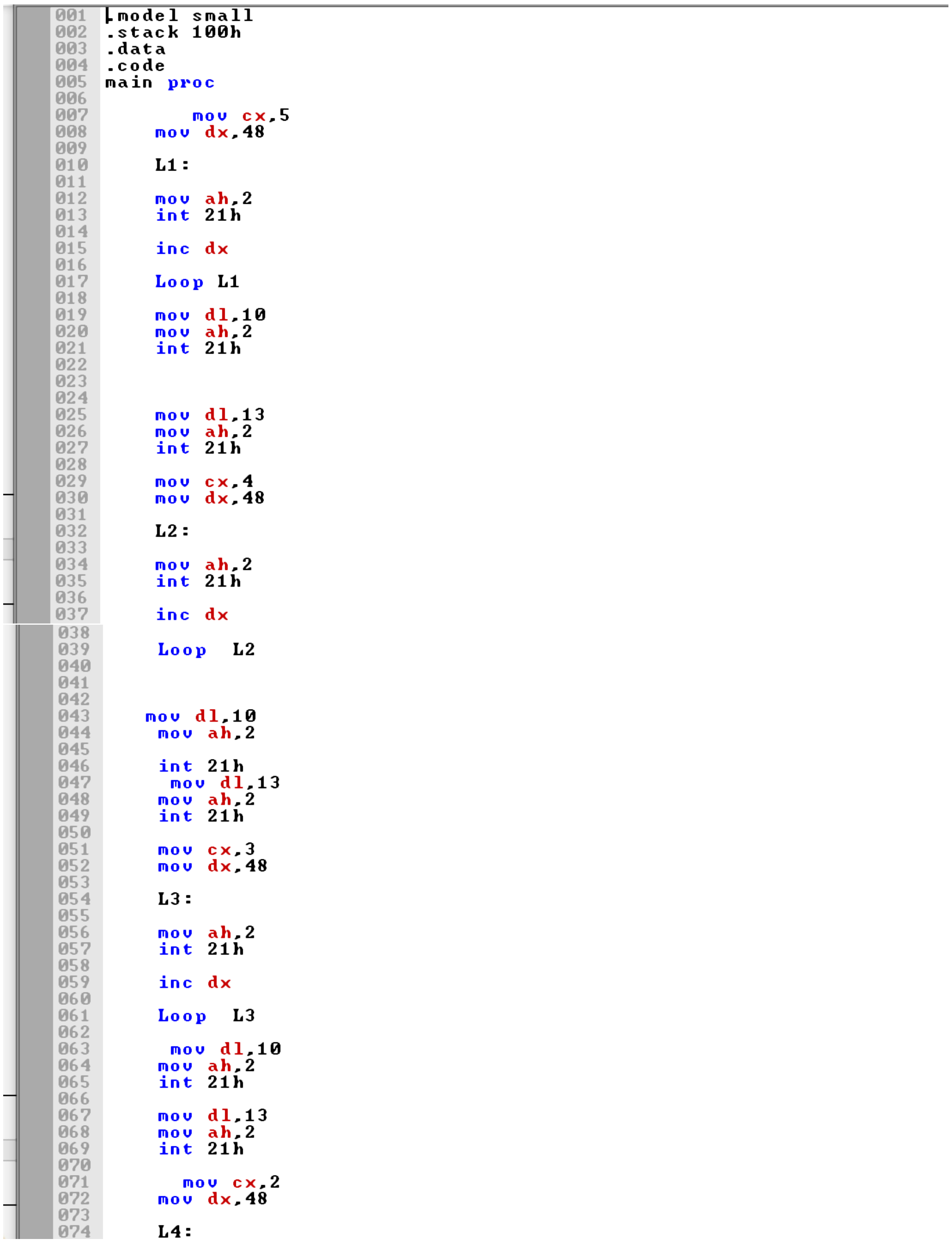
original sour...

```
main proc

mov cx,26
mov dx,65

L1:

mov ah,2
int 21h

inc dx

Loop L1



mov cx,26
mov dx,90

L2:

mov ah,2
int 21h

dec dx

Loop L2
mov ah,4ch
int 21h
```

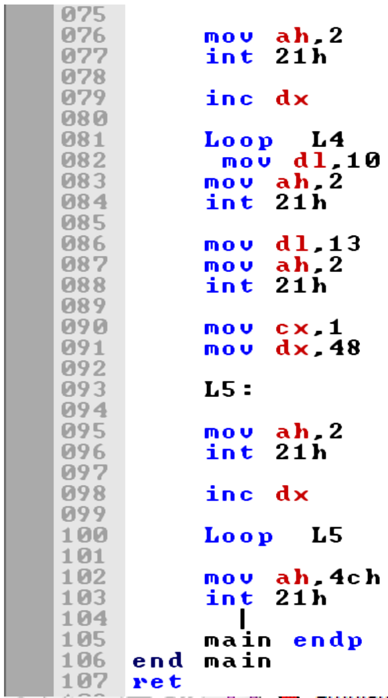
emulator screen (80x25 chars)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZZYXWVUTSRQPONMLKJIHGFEDCBA
```

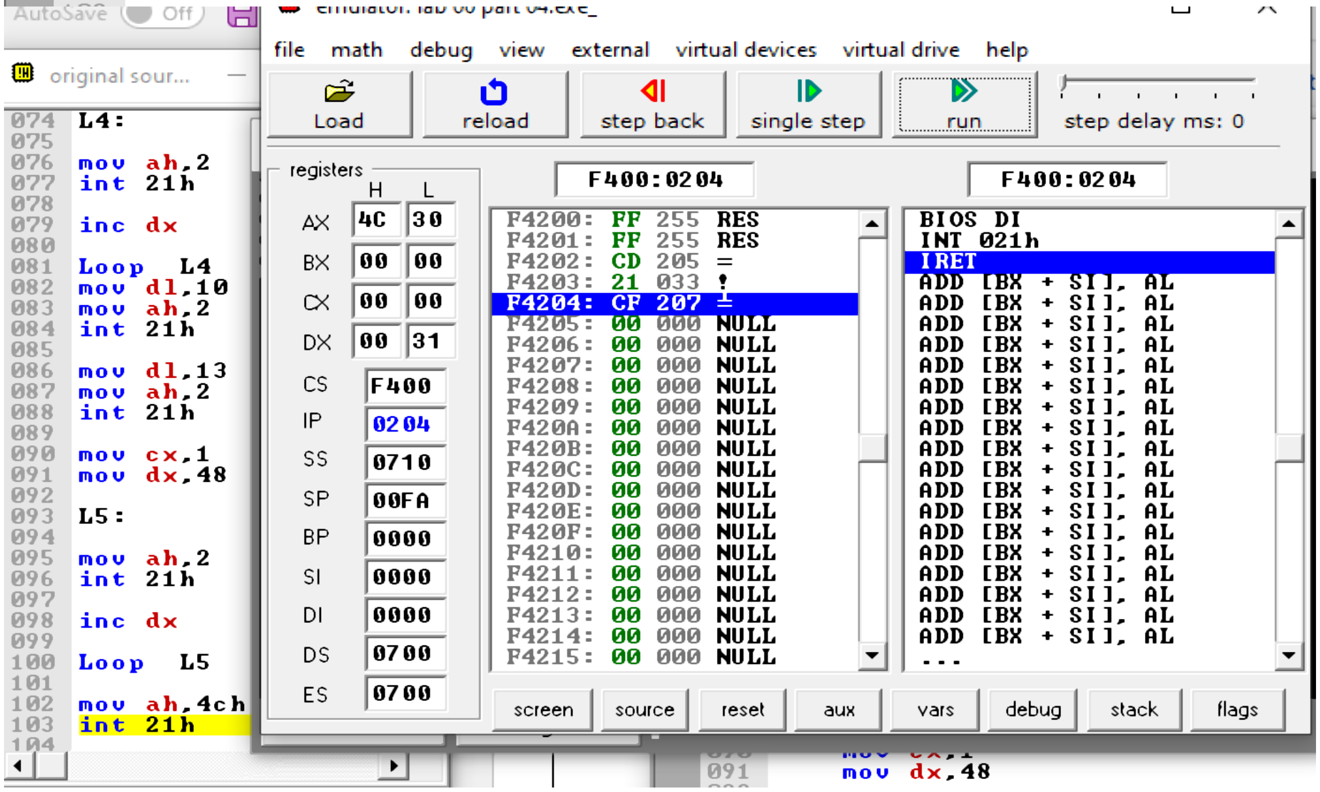
Task#03: Write a program to print Pyramid of numbers character using loop



```
.model small
.stack 100h
.data
.code
main proc

       mov cx,5
    mov dx,48

    L1:

    mov ah,2
    int 21h

    inc dx

    Loop L1

    mov dl,10
    mov ah,2
    int 21h


    mov dl,13
    mov ah,2
    int 21h

    mov cx,4
    mov dx,48

    L2:

    mov ah,2
    int 21h

    inc dx

    Loop  L2



   mov dl,10
    mov ah,2

    int 21h
     mov dl,13
    mov ah,2
    int 21h

    mov cx,3
    mov dx,48

    L3:

    mov ah,2
    int 21h

    inc dx

    Loop  L3

     mov dl,10
    mov ah,2
    int 21h

    mov dl,13
    mov ah,2
    int 21h

      mov cx,2
    mov dx,48

    L4:
```



```
        mov ah,2
        int 21h

        inc dx

        Loop  L4
         mov dl,10
        mov ah,2
        int 21h

        mov dl,13
        mov ah,2
        int 21h

        mov cx,1
        mov dx,48

        L5:

        mov ah,2
        int 21h

        inc dx

        Loop  L5

        mov ah,4ch
        int 21h

        main endp
    end main
    ret
```



AutoSave Off
original sour...
emulator: lab 06 part 04.exe_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
screen source reset aux vars debug stack flags

```
L4:

mov ah,2
int 21h

inc dx

Loop  L4
mov dl,10
mov ah,2
int 21h

mov dl,13
mov ah,2
int 21h

mov cx,1
mov dx,48

L5:

mov ah,2
int 21h

inc dx

Loop  L5

mov ah,4ch
int 21h
```

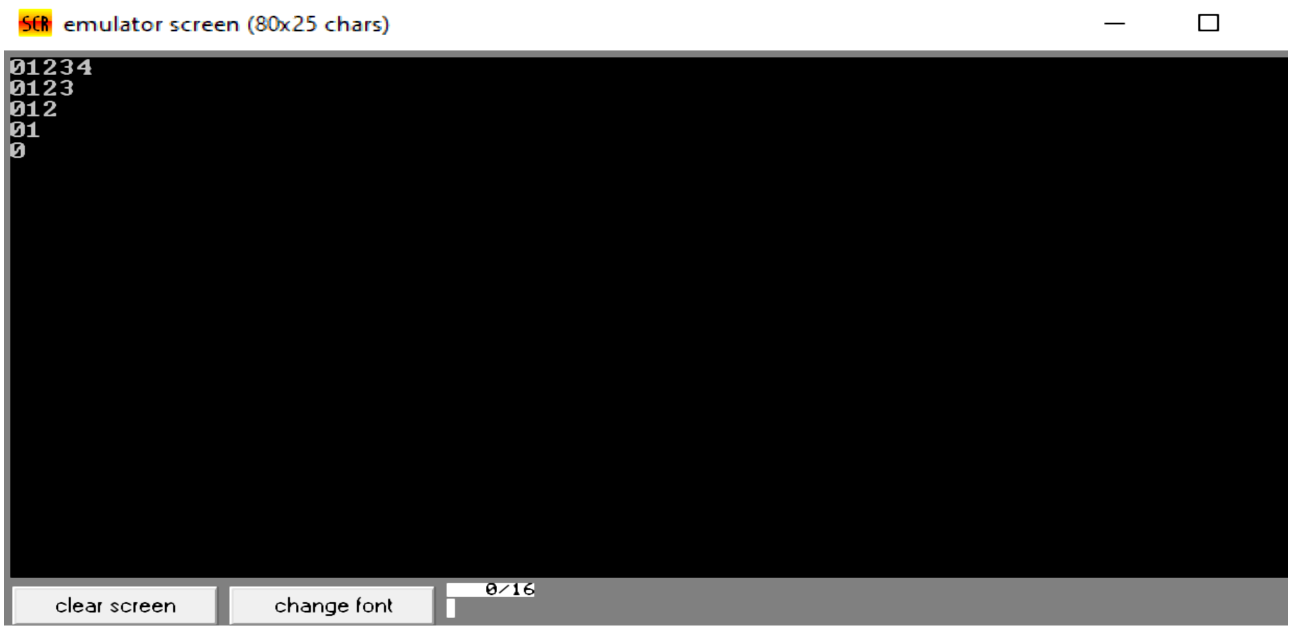
registers

	H	L
AX	4C	30
BX	00	00
CX	00	00
DX	00	31
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
F400:0204

F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
```

```
F400:0204

BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

```
mov dx,48
```



```
01234
0123
012
01
0
```