Name:	Pankhania Aanandi R.
Roll No:	IT081
Batch:	I1

Experiment 2

1. Write an assembly language program for subtraction of two 16-bit numbers.

Rules for Operands: 1st number=your roll no. of sem-5, 2nd number=reverse of your roll no.

e.g. roll no.=IT108 so, 16-bit 1st number = 0108h, 2nd number=8010h.

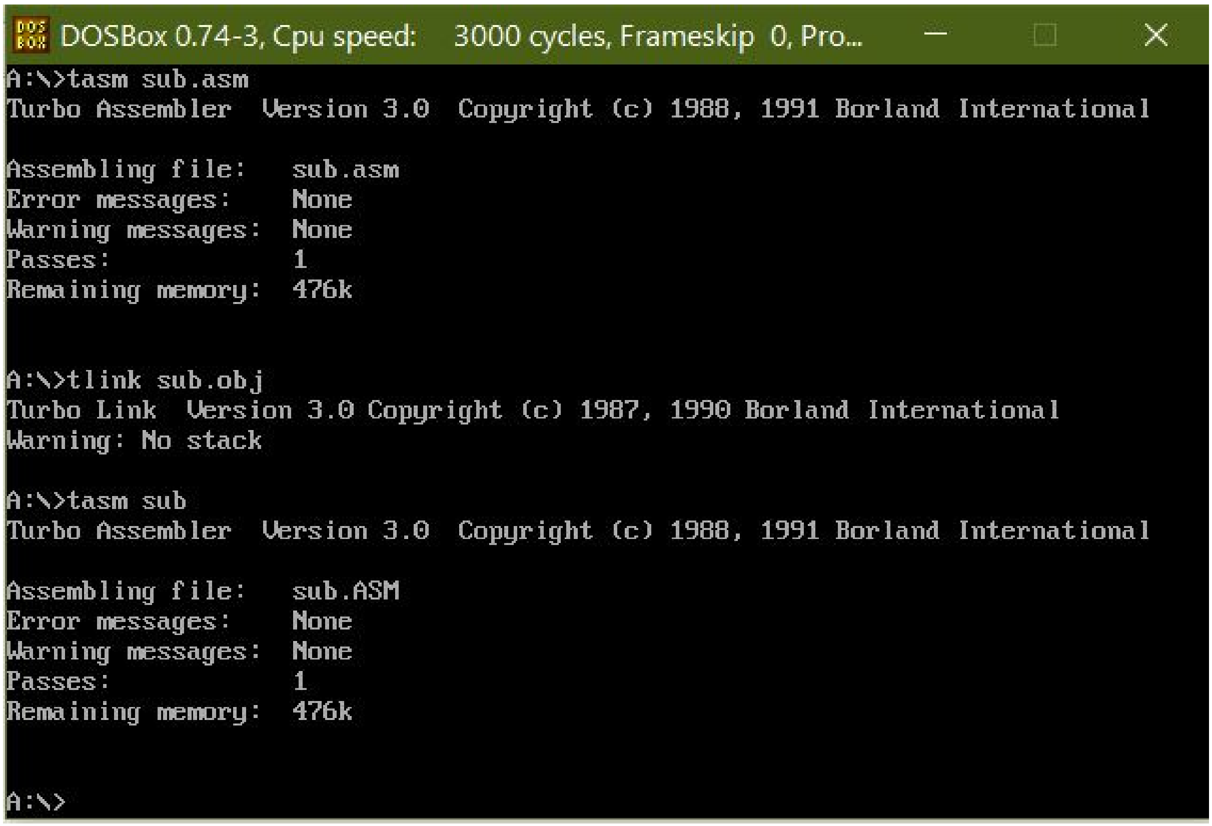
e.g. for repeater student ID=18ITUOS103, 1st number=1803h, 2nd number=3081h.

Write your code here:

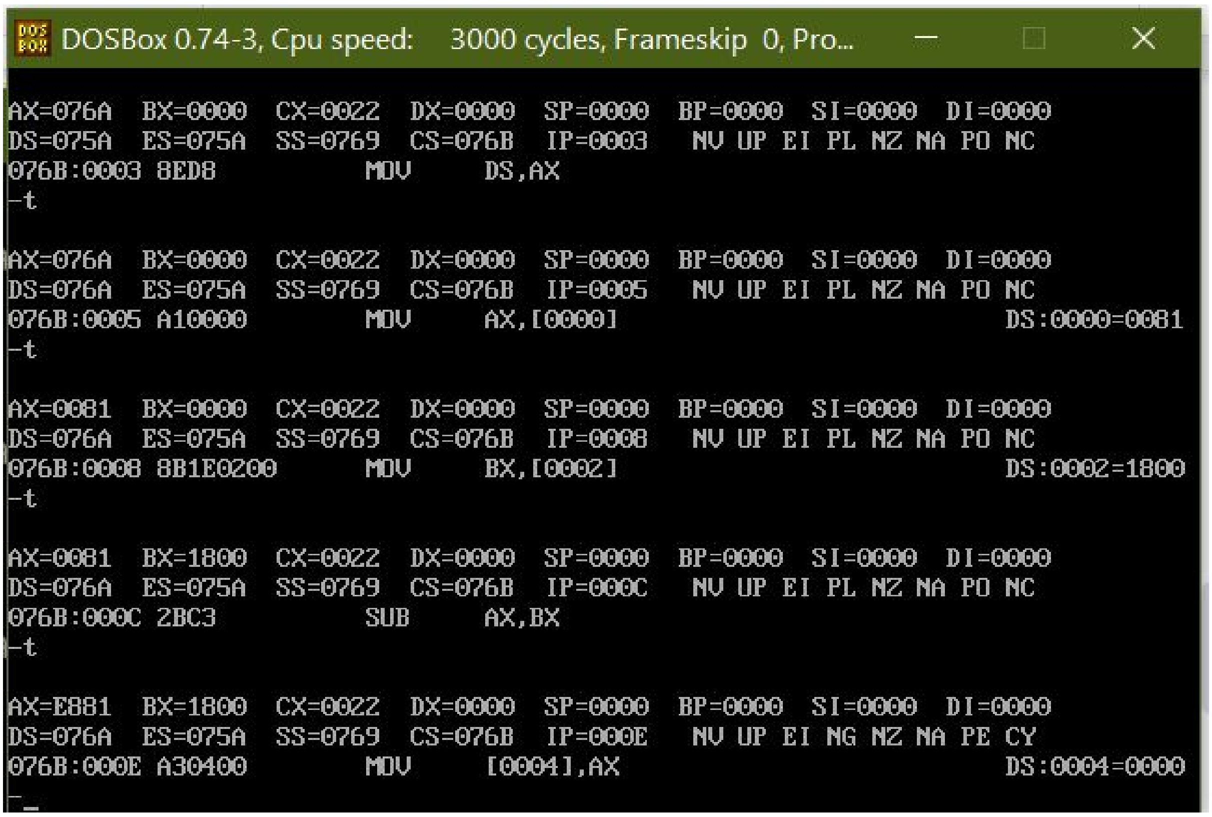
```
data_here segment
            a dw 0081h
            b dw 1800h
            c dw ?
data_here ends

code_here segment
            assume cs:code_here,ds:data_here
            start:
                    mov ax,data_here
                    mov ds,ax
                    mov ax,a   ;ax=0081h   ; 129d
                    mov bx,b   ;bx=1800h    ;6144d
                    sub  ax,bx      ;ax=ax-bx     ;129d-6144d   (0081h-1800h)=   -6015d(2's complement of E881h)
                    mov c,ax    ;c=ax
                    int 3h

code_here ends
end start
```

Compilation /Running and Debugging steps:

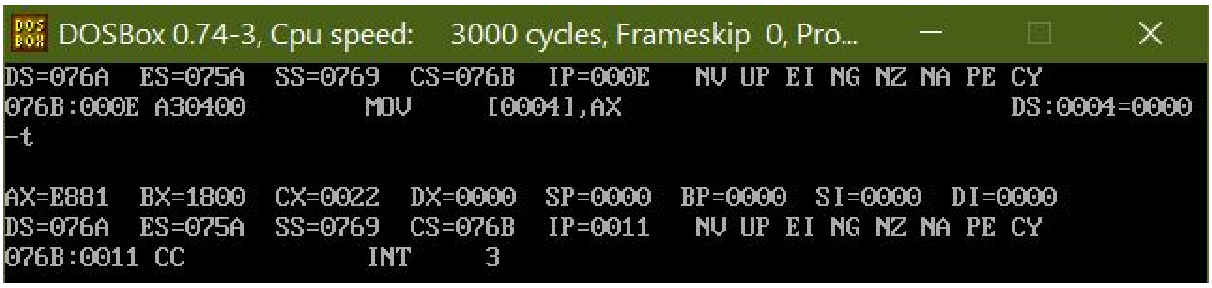
(As given in lab manual as an example of multiplication program on page no:5 of lab manual)


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
A:\>tasm sub.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   sub.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


A:\>tlink sub.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm sub
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   sub.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


A:\>
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
AX=076A  BX=0000  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076B  IP=0003   NV UP EI PL NZ NA PO NC
076B:0003 8ED8          MOV     DS,AX
-t

AX=076A  BX=0000  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0005   NV UP EI PL NZ NA PO NC
076B:0005 A10000        MOV     AX,[0000]                          DS:0000=0081
-t

AX=0081  BX=0000  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0008   NV UP EI PL NZ NA PO NC
076B:0008 8B1E0200      MOV     BX,[0002]                          DS:0002=1800
-t

AX=0081  BX=1800  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=000C   NV UP EI PL NZ NA PO NC
076B:000C 2BC3          SUB     AX,BX
-t

AX=E881  BX=1800  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=000E   NV UP EI NG NZ NA PE CY
076B:000E A30400        MOV     [0004],AX                          DS:0004=0000
-_
```

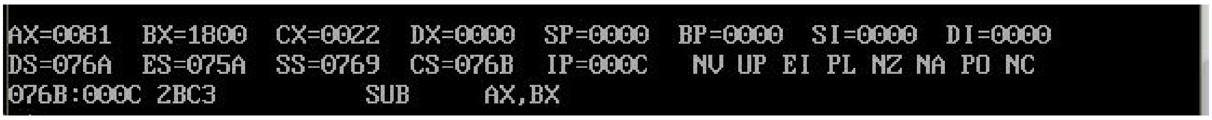

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
DS=076A  ES=075A  SS=0769  CS=076B  IP=000E   NV UP EI NG NZ NA PE CY
076B:000E A30400        MOV     [0004],AX                        DS:0004=0000
-t

AX=E881  BX=1800  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0011   NV UP EI NG NZ NA PE CY
076B:0011 CC            INT     3
```

Output:

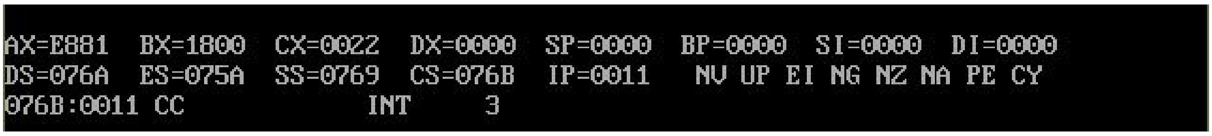
Screenshots of internal register contents before execution and after execution.

Before execution:



```
AX=0081  BX=1800  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=000C   NV UP EI PL NZ NA PO NC
076B:000C 2BC3          SUB     AX,BX
```

After execution:



```
AX=E881  BX=1800  CX=0022  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0011   NV UP EI NG NZ NA PE CY
076B:0011 CC            INT     3
```

2. **Write an assembly language program to perform scalar multiplication of an array of five unsigned bytes.**
   Rules for Operands: Array elements are 10h,15h,20h,25h,30h (you can take any values). Multiply each element by the last digit of your roll no./ (repeater student – student id) i.e. IT067 so, multiply array elements by 7 and store result in another array.

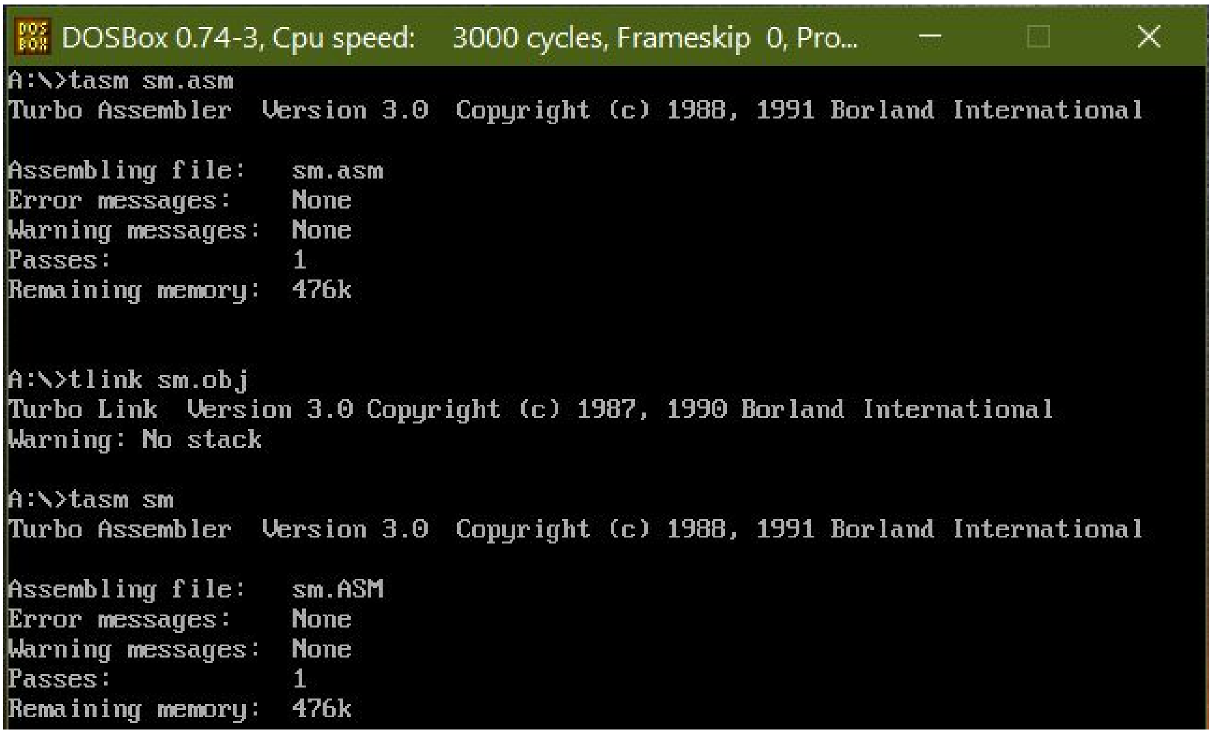
Write your code here:

```
data_here segment
            arr db 1h,2h,3h,14h,25h  ;created an array
            ld db 1h  ;roll no.'s(IT081) last digit---1
            ar dw 5 dup(?)  ;another array
data_here ends
code_here segment
            assume cs:code_here,ds:data_here
            start:    mov ax,data_here
                      mov ds,ax
```

```
                mov cl, 5   ;count value 5
                mov bl, ld
                mov DI, 0
                mov ah,0
        l1: mov al, arr[DI] ;loop l1 start
                mul bl
                mov ar[DI], al
                inc DI
                dec cl    ;count--
                jnz l1    ;when count 0 loop will end...
                int 21h
                int 3h
code_here ends
end start
```

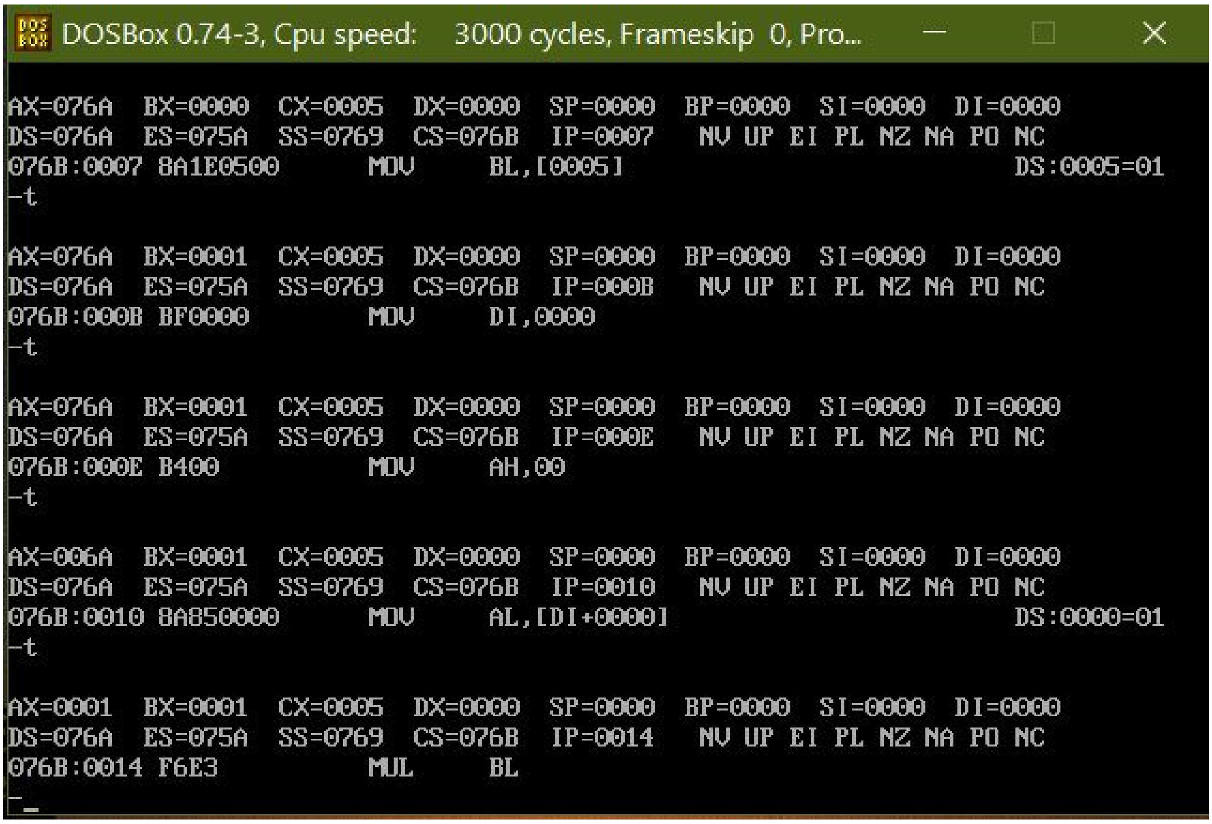
Compilation /Running and Debugging steps:

(As given in the lab manual as an example of multiplication program on page no:5 of lab manual)

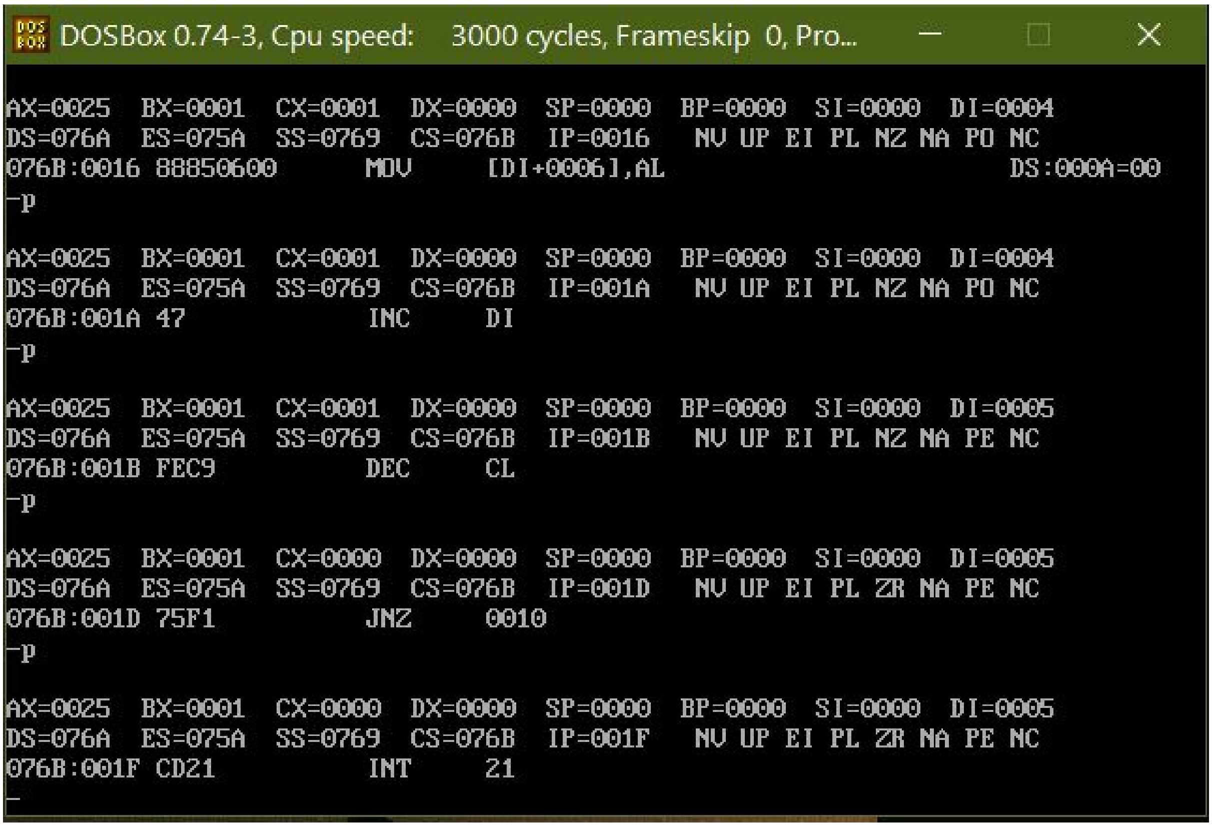


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
A:\>tasm sm.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   sm.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


A:\>tlink sm.obj
Turbo Link  Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm sm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   sm.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
AX=076A  BX=0000  CX=0005  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0007   NV UP EI PL NZ NA PO NC
076B:0007 8A1E0500      MOV     BL,[0005]                         DS:0005=01
-t

AX=076A  BX=0001  CX=0005  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=000B   NV UP EI PL NZ NA PO NC
076B:000B BF0000        MOV     DI,0000
-t

AX=076A  BX=0001  CX=0005  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=000E   NV UP EI PL NZ NA PO NC
076B:000E B400          MOV     AH,00
-t

AX=006A  BX=0001  CX=0005  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0010   NV UP EI PL NZ NA PO NC
076B:0010 8A850000      MOV     AL,[DI+0000]                      DS:0000=01
-t

AX=0001  BX=0001  CX=0005  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0014   NV UP EI PL NZ NA PO NC
076B:0014 F6E3          MUL     BL
-_
```

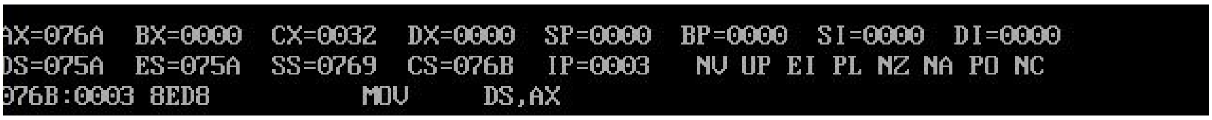

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...

```
AX=0025  BX=0001  CX=0001  DX=0000  SP=0000  BP=0000  SI=0000  DI=0004
DS=076A  ES=075A  SS=0769  CS=076B  IP=0016   NV UP EI PL NZ NA PO NC
076B:0016 88850600      MOV     [DI+0006],AL                      DS:000A=00
-p

AX=0025  BX=0001  CX=0001  DX=0000  SP=0000  BP=0000  SI=0000  DI=0004
DS=076A  ES=075A  SS=0769  CS=076B  IP=001A   NV UP EI PL NZ NA PO NC
076B:001A 47            INC     DI
-p

AX=0025  BX=0001  CX=0001  DX=0000  SP=0000  BP=0000  SI=0000  DI=0005
DS=076A  ES=075A  SS=0769  CS=076B  IP=001B   NV UP EI PL NZ NA PE NC
076B:001B FEC9          DEC     CL
-p

AX=0025  BX=0001  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0005
DS=076A  ES=075A  SS=0769  CS=076B  IP=001D   NV UP EI PL ZR NA PE NC
076B:001D 75F1          JNZ     0010
-p

AX=0025  BX=0001  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0005
DS=076A  ES=075A  SS=0769  CS=076B  IP=001F   NV UP EI PL ZR NA PE NC
076B:001F CD21          INT     21
-
```

Output:

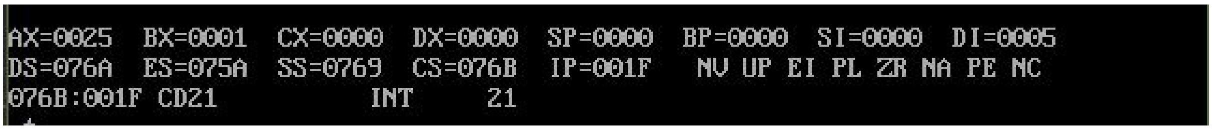
Screenshots of internal register contents before execution and after execution.

Before Execution:



```
AX=076A  BX=0000  CX=0032  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076B  IP=0003   NV UP EI PL NZ NA PO NC
076B:0003 8ED8          MOV     DS,AX
```

After Execution:



```
AX=0025  BX=0001  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0005
DS=076A  ES=075A  SS=0769  CS=076B  IP=001F   NV UP EI PL ZR NA PE NC
076B:001F CD21          INT     21
```