U18CO018
Shubham Shekhaliya
MIT
Assignment-9

1-> Program to multiply signed 16-bit numbers.

Code:-

```
model small
.8086
.data
a dw 0004H
b dw -0002H
c dw 0
d dw 0
.code
mov ax, @data
mov ds, ax
mov ax, a
mov bx, b
imul bx
mov c, ax
mov d, dx
mov ax, 4C00H
int 21h
end
```

Output:-

```
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 A10A00        MOV     AX,[000A]
076A:0008 8B1E0C00      MOV     BX,[000C]
076A:000C F7EB          IMUL    BX
076A:000E A30E00        MOV     [000E],AX
076A:0011 89161000      MOV     [0010],DX
076A:0015 B8004C        MOV     AX,4C00
076A:0018 CD21          INT     21
076A:001A 0400          ADD     AL,00
076A:001C FEFF          ???     BH
076A:001E 0000          ADD     [BX+SI],AL
-g

Program terminated normally
-d 076B:0000
076B:0000  00 89 16 10 00 B8 00 4C-CD 21 04 00 FE FF F8 FF   .......L.!......
076B:0010  FF FF 00 80 0E 10 00 02-00 80 0E 00 00 00 00 81   ................
076B:0020  0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0070  FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF   ...&............
-_
```

2-> Program to multiply unsigned 16-bit numbers.

Code:-

```
model small
.8086

.data
a dw 0004H
b dw 0FFFEH
c dw 0
d dw 0

.code
mov ax, @data
mov ds, ax

mov ax, a
mov bx, b
mul bx

mov c, ax
mov d, dx


mov ax, 4C00H
int 21h
end
```

Output:-

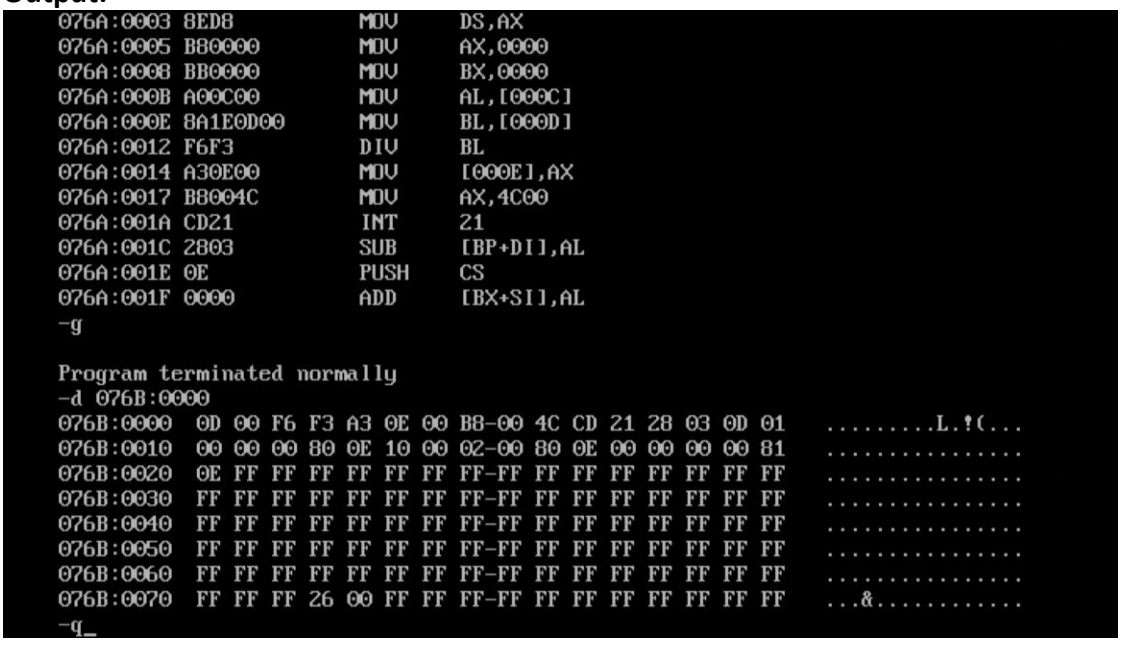
```
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 A10A00        MOV     AX,[000A]
076A:0008 8B1E0C00      MOV     BX,[000C]
076A:000C F7E3          MUL     BX
076A:000E A30E00        MOV     [000E],AX
076A:0011 89161000      MOV     [0010],DX
076A:0015 B8004C        MOV     AX,4C00
076A:0018 CD21          INT     21
076A:001A 0400          ADD     AL,00
076A:001C FEFF          ???     BH
076A:001E 0000          ADD     [BX+SI],AL
-g

Program terminated normally
-d 076B:0000
076B:0000  00 89 16 10 00 B8 00 4C-CD 21 04 00 FE FF F8 FF   .......L.!......
076B:0010  03 00 00 80 0E 10 00 02-00 80 0E 00 00 00 00 81   ................
076B:0020  0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0070  FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF   ...&............
-
```

3-> Program for division of unsigned 8-bit numbers.

Code:-

```
model small
.8086

.data
a db 28H
b db 03H
c dw ?

.code
mov ax, @data
mov ds, ax

mov ax,0000H
mov bx,0000H

mov al,a
mov bl,b
div bl

mov c,ax

mov ax, 4C00H
int 21h
end
```

Output:-

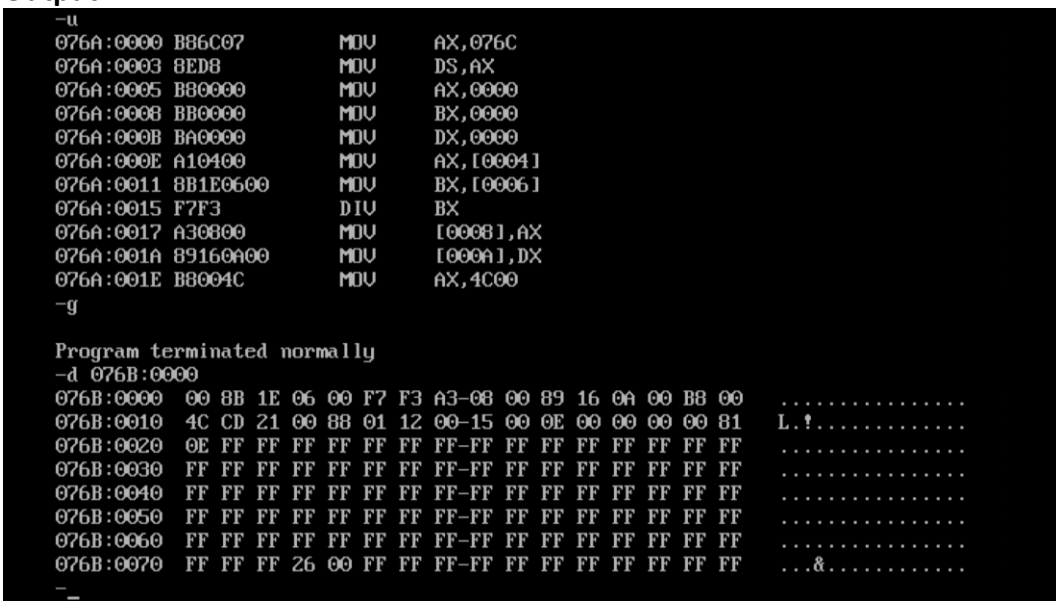


4-> Program for division of unsigned 16-bit numbers.

Code:-

```
model small
.8086

.data
a dw 0188H
b dw 0012H
c dw ?
d dw ?

.code
mov ax, @data
mov ds, ax
mov ax,0000H
mov bx,0000H
mov dx,0000H
mov ax,a
mov bx,b
div bx
mov c,ax
mov d,dx

mov ax, 4C00H
int 21h
end
```

Output:-



5-> Program for division of signed 8-bit numbers.

Code:-

```
model small
.8086

.data
a db 28H
b db -03H
c dw ?

.code
mov ax, @data
mov ds, ax

mov ax,0000H
mov bx,0000H

mov al,a
mov bl,b
idiv bl

mov c,ax

mov ax, 4C00H
int 21h
end
```

Output:-

```
076A:0003 8ED8          MOV     DS,AX
076A:0005 B80000        MOV     AX,0000
076A:0008 BB0000        MOV     BX,0000
076A:000B A00C00        MOV     AL,[000C]
076A:000E 8A1E0D00      MOV     BL,[000D]
076A:0012 F6FB          IDIV    BL
076A:0014 A30E00        MOV     [000E],AX
076A:0017 B8004C        MOV     AX,4C00
076A:001A CD21          INT     21
076A:001C 28FD          SUB     CH,BH
076A:001E 0E            PUSH    CS
076A:001F 0000          ADD     [BX+SI],AL
-g

Program terminated normally
-d 076B:0000
076B:0000  0D 00 F6 FB A3 0E 00 B8-00 4C CD 21 28 FD F3 01   .........L.!(...
076B:0010  00 00 00 80 0E 10 00 02-00 80 0E 00 00 00 00 81   ................
076B:0020  0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0070  FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF   ...&............
-
```

6-> Program for division of signed 16-bit numbers.

Code:-

```
model small
.8086
.data
a dw 0188H
b dw -0012H
c dw ?
d dw ?

.code
mov ax, @data
mov ds, ax
mov ax,0000H
mov bx,0000H
mov dx,0000H

mov ax,a
mov bx,b
idiv bx
mov c,ax
mov d,dx

mov ax, 4C00H
int 21h
end
```

Output:-

```
-u
076A:0000 B86C07        MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 B80000        MOV     AX,0000
076A:0008 BB0000        MOV     BX,0000
076A:000B BA0000        MOV     DX,0000
076A:000E A10400        MOV     AX,[0004]
076A:0011 8B1E0600      MOV     BX,[0006]
076A:0015 F7FB          IDIV    BX
076A:0017 A30800        MOV     [0008],AX
076A:001A 89160A00      MOV     [000A],DX
076A:001E B8004C        MOV     AX,4C00
-g

Program terminated normally
-d 076B:0000
076B:0000  00 8B 1E 06 00 F7 FB A3-08 00 89 16 0A 00 B8 00   ................
076B:0010  4C CD 21 00 88 01 EE FF-EB FF 0E 00 00 00 00 81   L.!.............
076B:0020  0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF   ................
076B:0070  FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF   ...&............
-_
```

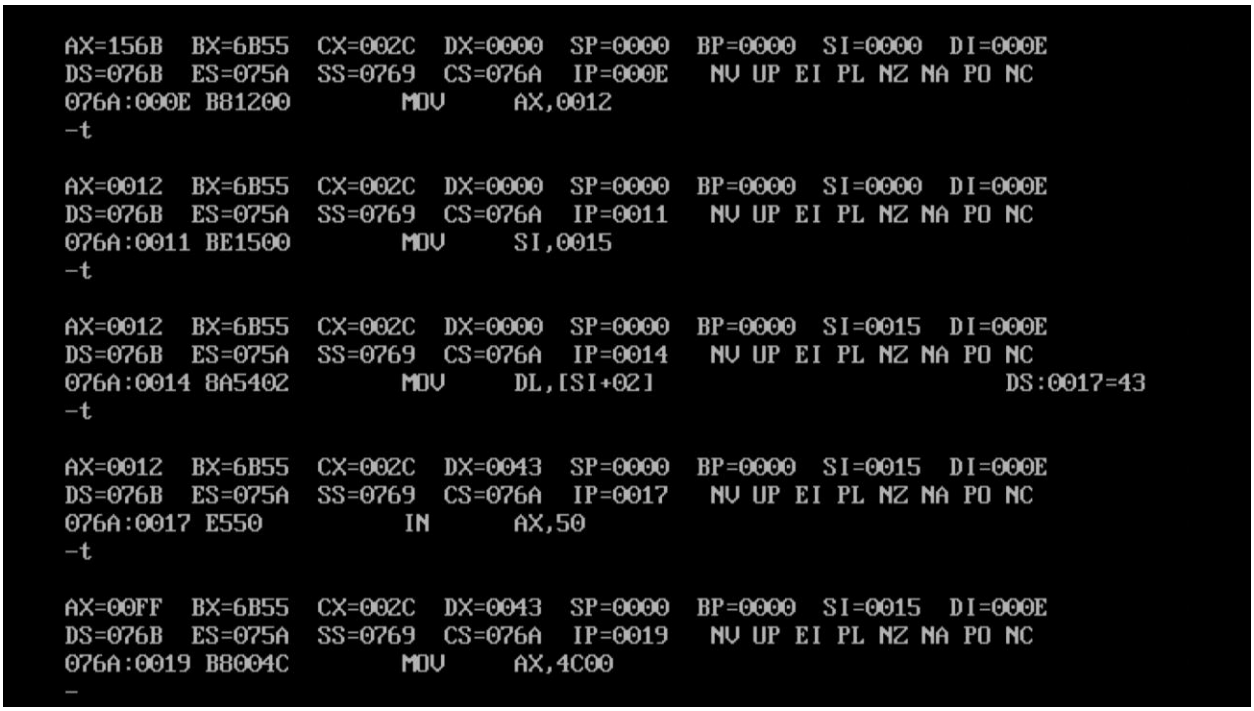
7-> Program for data transfer using different addressing modes.

Code:-

```
model small
.8086

.data
s1 db "UVWXYZ$"
s2 db "ABCDEF$"

.code
mov ax, @data
mov ds, ax

mov ah,15h
mov bh,al
mov di,offset s1
mov bl,[di]
mov ax,[0012h]
mov si, offset s2
mov dl,[si+2]
in ax , 50H

mov ax, 4C00H
int 21h
end
```

Output :-

```
AX=076B  BX=0000  CX=002C  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-t

AX=076B  BX=0000  CX=002C  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076B  ES=075A  SS=0769  CS=076A  IP=0005   NV UP EI PL NZ NA PO NC
076A:0005 B415          MOV     AH,15
-t

AX=156B  BX=0000  CX=002C  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076B  ES=075A  SS=0769  CS=076A  IP=0007   NV UP EI PL NZ NA PO NC
076A:0007 8AF8          MOV     BH,AL
-t

AX=156B  BX=6B00  CX=002C  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076B  ES=075A  SS=0769  CS=076A  IP=0009   NV UP EI PL NZ NA PO NC
076A:0009 BF0E00        MOV     DI,000E
-t

AX=156B  BX=6B00  CX=002C  DX=0000  SP=0000  BP=0000  SI=0000  DI=000E
DS=076B  ES=075A  SS=0769  CS=076A  IP=000C   NV UP EI PL NZ NA PO NC
076A:000C 8A1D          MOV     BL,[DI]                         DS:000E=55
-
```



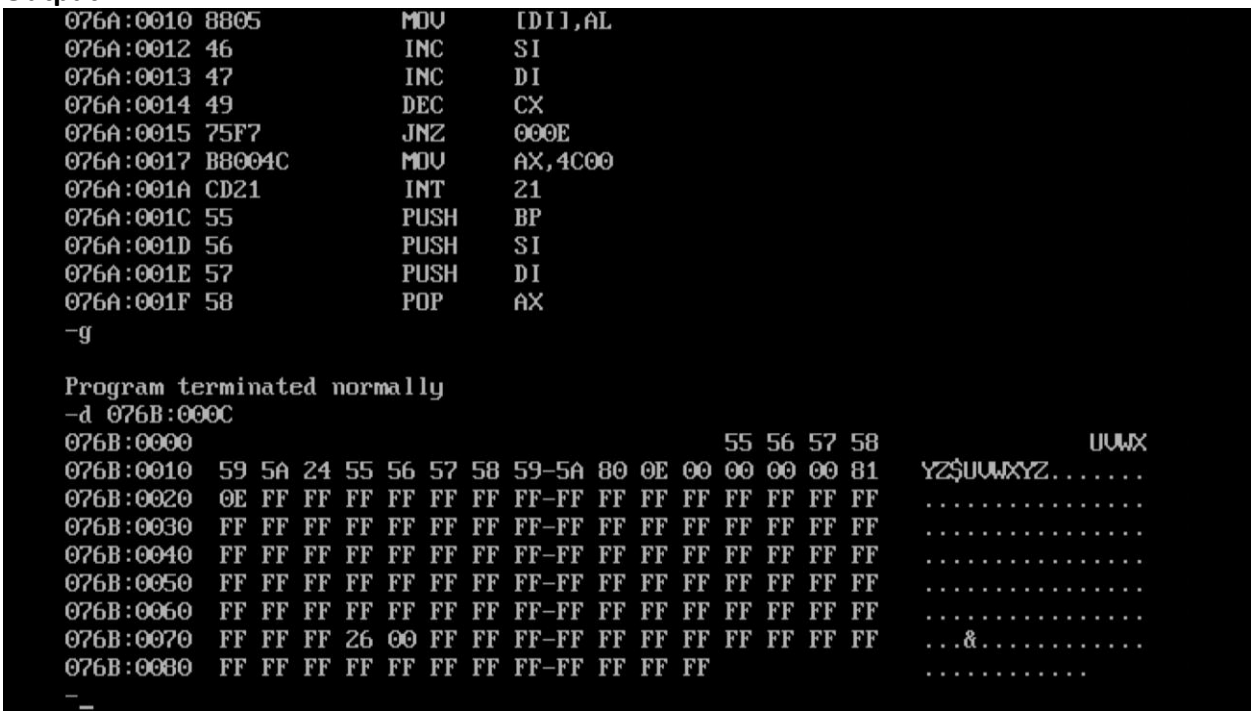
8-> Program to move data from source to destination using indirect addressing mode (Block Move without overlap).

Code:-

```
model small
.8086

.data
s1 db "UVWXYZ$"
s2 db 6 dup(0)

.code
mov ax, @data
mov ds, ax

mov si,offset s1
mov di,offset s2
mov cx,0006h

up: mov al,[si]
   mov [di],al
   inc si
   inc di
   dec cx
   jnz up
```

mov ax, 4C00H
int 21h
end

Output :-



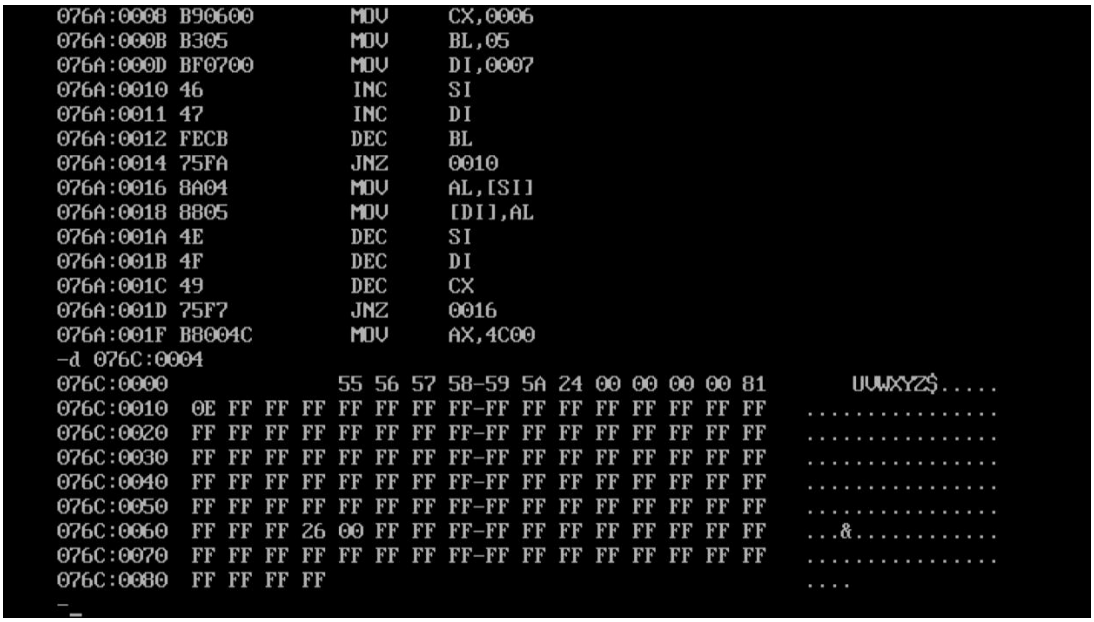
9-> Program to move a block of data from source to destination (With overlap in either direction.

Code:-

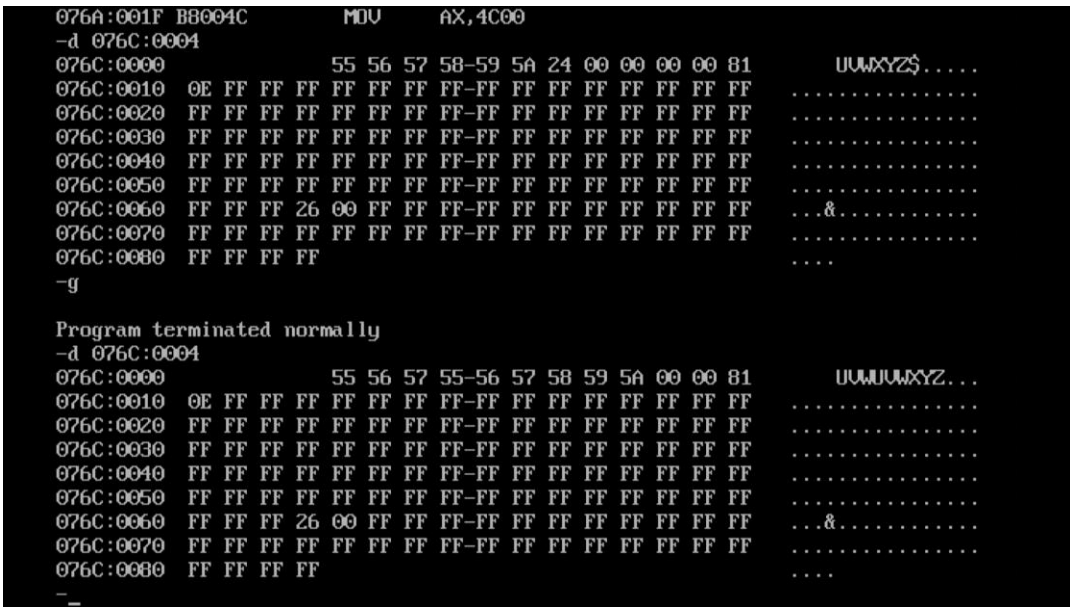
```
model small
.8086
.data
s1 db "UVWXYZ$"

.code
mov ax, @data
mov ds, ax
mov si,offset s1
mov cx,0006h
mov bl,05h
mov di,offset [s1+3]
  up: inc si
    inc di
    dec bl
    jnz up
```

```
   go: mov al,[si]
       mov [di],al
       dec si
       dec di
       dec cx
       jnz go
mov ax, 4C00H
int 21h
end
```

Before :-



After :-

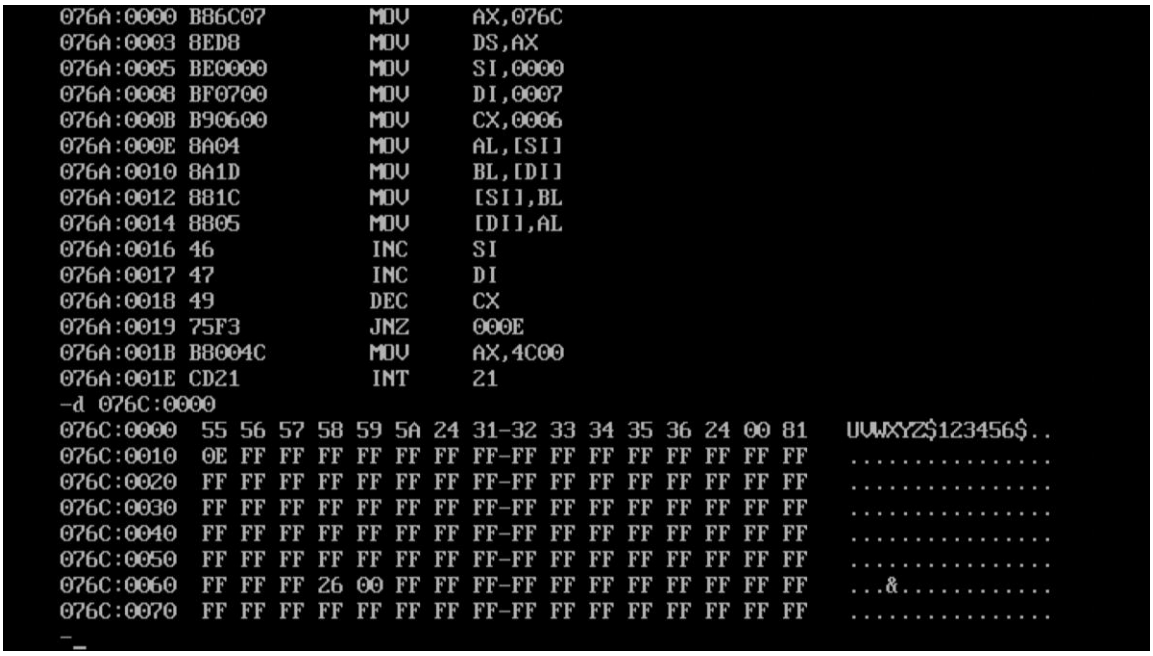


10-> Program to interchange two blocks of data.

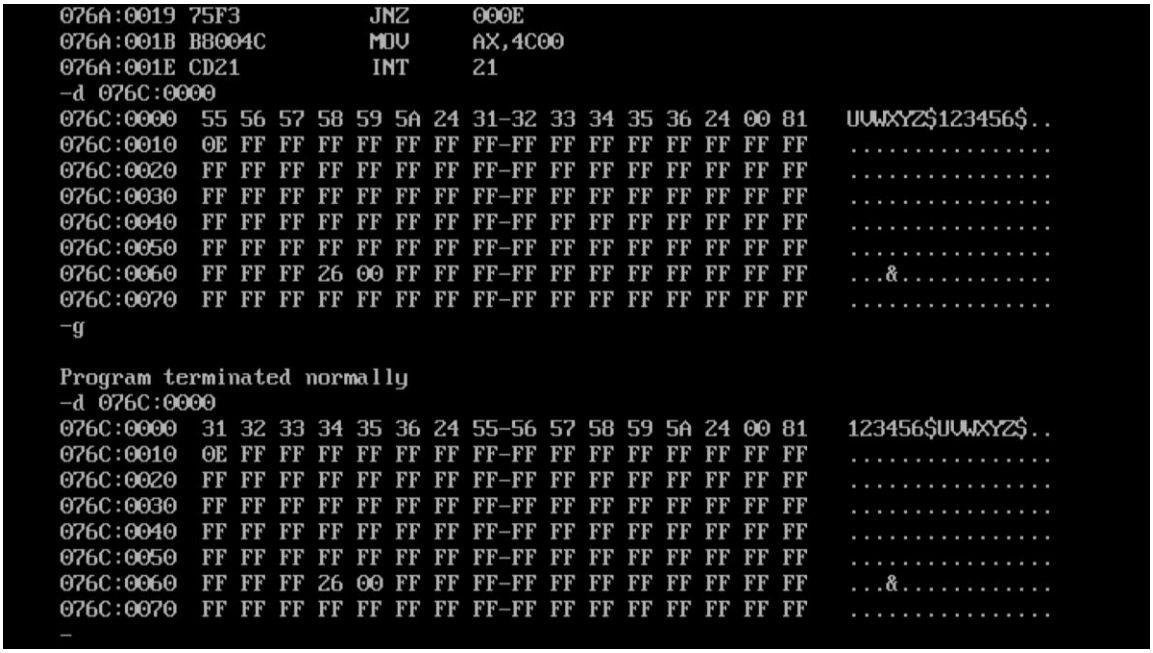
Code:-

```
model small
.8086
.data
s1 db "UVWXYZ$"
s2 db "123456$"

.code
mov ax, @data
mov ds, ax
mov si,offset s1
mov di,offset s2
mov cx,0006h
up: mov al,[si]
   mov bl,[di]
   mov [si],bl
   mov [di],al
   inc si
   inc di
   dec cx
   jnz up
mov ax, 4C00H
int 21h
end
```

Before:-



After:-


076A:0019 75F3 JNZ 000E
076A:001B B8004C MOV AX,4C00
076A:001E CD21 INT 21
-d 076C:0000
076C:0000 55 56 57 58 59 5A 24 31-32 33 34 35 36 24 00 81 UVWXYZ123456..
076C:0010 0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0020 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF ...&............
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
-g
Program terminated normally
-d 076C:0000
076C:0000 31 32 33 34 35 36 24 55-56 57 58 59 5A 24 00 81 123456$UVWXYZ$..
076C:0010 0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0020 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF ...&............
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ....
-