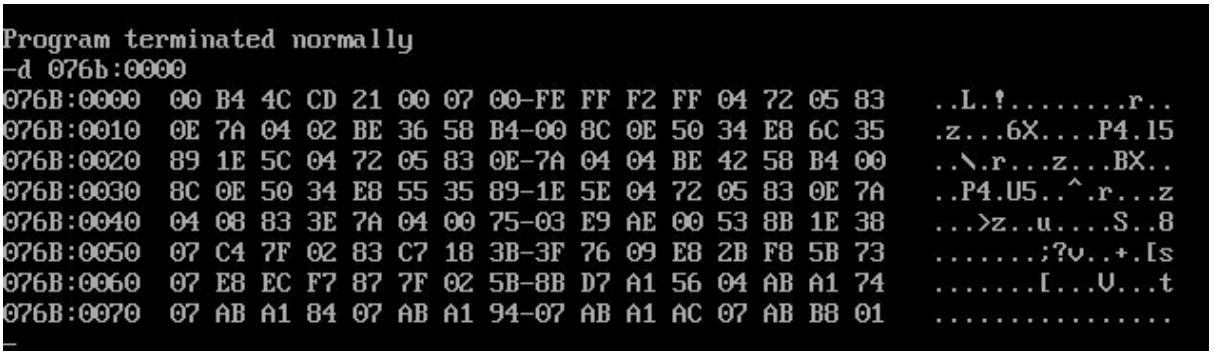
MIT LAB 9

U18CO021: SAHIL BONDRE

1. Program to multiply signed 16-bit numbers

```
.model small

.8086
.data
  a dw 7
  b dw -2
  prod dw ?

.code
  mov ax, @data
  mov ds, ax

  mov ax, a
  mov bx, b
  imul bx
  mov prod, ax

  mov ah, 4ch
  int 21h
end
```

```
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 A10600        MOV     AX,[0006]
076A:0008 8B1E0800      MOV     BX,[0008]
076A:000C F7EB          IMUL    BX
076A:000E A30A00        MOV     [000A],AX
076A:0011 B44C          MOV     AH,4C
076A:0013 CD21          INT     21
076A:0015 0007          ADD     [BX],AL
076A:0017 00FE          ADD     DH,BH
076A:0019 FFF2          PUSH    DX
076A:001B FF04          INC     WORD PTR [SI]
076A:001D 7205          JB      0024
076A:001F 830E7A0402    OR      WORD PTR [047A],+02
```



[076B:0000] = F2FEH = (-14) = 7 * -2

2. Program to multiply unsigned 16-bit numbers

```
.model small

.8086
.data
  a dw 7
  b dw 2
  prod dw ?

.code
  mov ax, @data
  mov ds, ax

  mov ax, a
  mov bx, b
  mul bx
  mov prod, ax

  mov ah, 4ch
  int 21h
end
```

```
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 A10600        MOV     AX,[0006]
076A:0008 8B1E0800      MOV     BX,[0008]
076A:000C F7E3          MUL     BX
076A:000E A30A00        MOV     [000A],AX
076A:0011 B44C          MOV     AH,4C
076A:0013 CD21          INT     21
076A:0015 0007          ADD     [BX],AL
076A:0017 0002          ADD     [BP+SI],AL
076A:0019 001E5A04      ADD     [045A],BL
076A:001D 7205          JB      0024
076A:001F 830E7A0402    OR      WORD PTR [047A],+02
```

```
Program terminated normally
-d 076b:0000
076B:0000  00 B4 4C CD 21 00 07 00-02 00 0E 00 04 72 05 83   ..L.!........r..
076B:0010  0E 7A 04 02 BE 36 58 B4-00 8C 0E 50 34 E8 6C 35   .z...6X....P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
```

[076B:0000] = 000E = 14 = 2 * 7

3. Program for division of unsigned 8-bit numbers

```
model small

.8086
.data
quo db ?
rem db ?

.code
  mov ax,@data
  mov ds,ax
  mov ax, 0020h
  mov bl, 10h
  div bl
  mov quo, al
  mov rem, ah
  mov ax, 4c00h
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q3.EXE
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 B82000        MOV     AX,0020
076A:0008 B310          MOV     BL,10
076A:000A F6F3          DIV     BL
076A:000C A20800        MOV     [0008],AL
076A:000F 88260900      MOV     [0009],AH
076A:0013 B8004C        MOV     AX,4C00
076A:0016 CD21          INT     21
076A:0018 35891E        XOR     AX,1E89
076A:001B 5A            POP     DX
076A:001C 0472          ADD     AL,72
076A:001E 05830E        ADD     AX,0E83
-g

Program terminated normally
-
```

```
Program terminated normally
-d 076b:0000
076B:0000  26 09 00 B8 00 4C CD 21-02 00 1E 5A 04 72 05 83   &....L.!...Z.r..
076B:0010  0E 7A 04 02 BE 36 58 B4-00 8C 0E 50 34 E8 6C 35   .z...6X....P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
-
```

[076b:0008] = 20 / 10 = 2, [076b:0009] = 20 % 10 = 0

4. Program for division of unsigned 16-bit numbers

```
model small

.8086
.data
quo db ?
rem db ?

.code
mov ax,@data
mov ds,ax
mov ax, 1234H
mov bx, 22H
div bx
```

```
mov quo, al
mov rem, ah
mov ax, 4c00h
int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q4.EXE
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 B83412        MOV     AX,1234
076A:0008 BB2200        MOV     BX,0022
076A:000B F7F3          DIV     BX
076A:000D A20A00        MOV     [000A],AL
076A:0010 88260B00      MOV     [000B],AH
076A:0014 B8004C        MOV     AX,4C00
076A:0017 CD21          INT     21
076A:0019 89890004      MOV     [BX+DI+0400],CX
076A:001D 7205          JB      0024
076A:001F 830E7A0402    OR      WORD PTR [047A],+02
-g
```

```
-d 076b:0000
076B:0000  88 26 0B 00 B8 00 4C CD-21 89 89 00 04 72 05 83   .&....L.!....r..
076B:0010  0E 7A 04 02 BE 36 58 B4-00 8C 0E 50 34 E8 6C 35   .z...6X....P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
-
```

[076b:000A] = 89 = 1234H/22H

5. Program for division of signed 8-bit numbers

```
model small

.8086
.data
quo db ?
rem db ?

.code
  mov ax, @data
  mov ds, ax
  mov ax, -0010h
```

```
  mov bx, 0090h
  idiv bx
  mov quo, al
  mov rem, ah
  mov ax, 4c00h
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q5.EXE
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 B8F0FF        MOV     AX,FFF0
076A:0008 BB9000        MOV     BX,0090
076A:000B F7FB          IDIV    BX
076A:000D A20A00        MOV     [000A],AL
076A:0010 88260B00      MOV     [000B],AH
076A:0014 B8004C        MOV     AX,4C00
076A:0017 CD21          INT     21
076A:0019 891E5A04      MOV     [045A],BX
076A:001D 7205          JB      0024
076A:001F 830E7A0402    OR      WORD PTR [047A],+02
-g

Program terminated normally
```

```
Program terminated normally
-d 076b:0000
076B:0000  88 26 0B 00 B8 00 4C CD-21 89 C7 01 04 72 05 83   .&....L.!....r..
076B:0010  0E 7A 04 02 BE 36 58 B4-00 8C 0E 50 34 E8 6C 35   .z...6X....P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
-
```

6. Program for division of signed 16-bit numbers

```
model small

.8086
.data
quo db ?
rem db ?

.code
  mov ax,@data
  mov ds,ax
```

```
  mov ax, -0D10h
  mov bx, 00990h
  idiv bx
  mov quo, al
  mov rem, ah
  mov ax, 4c00h
  int 21h
end
```

```
-u
076A:0000 B86B07        MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 B8F0F2        MOV     AX,F2F0
076A:0008 BB9009        MOV     BX,0990
076A:000B F7FB          IDIV    BX
076A:000D A20A00        MOV     [000A],AL
076A:0010 88260B00      MOV     [000B],AH
076A:0014 B8004C        MOV     AX,4C00
076A:0017 CD21          INT     21
076A:0019 891E5A04      MOV     [045A],BX
076A:001D 7205          JB      0024
076A:001F 830E7A0402    OR      WORD PTR [047A],+02
-g

Program terminated normally
-_
```

7. Program for data transfer using different addressing modes

```
model small

.8086
.data
d1 db "string$"
d2 db "data$"

.code
  mov ax, @data
  mov ds, ax
  mov ah, 02h
  mov bh, al
  mov di, offset d2
  mov bl, [di]
  mov ax, [0011h]
  mov si, offset d1
  mov dl, [si + 4]
  mov ax, 4c00h
```

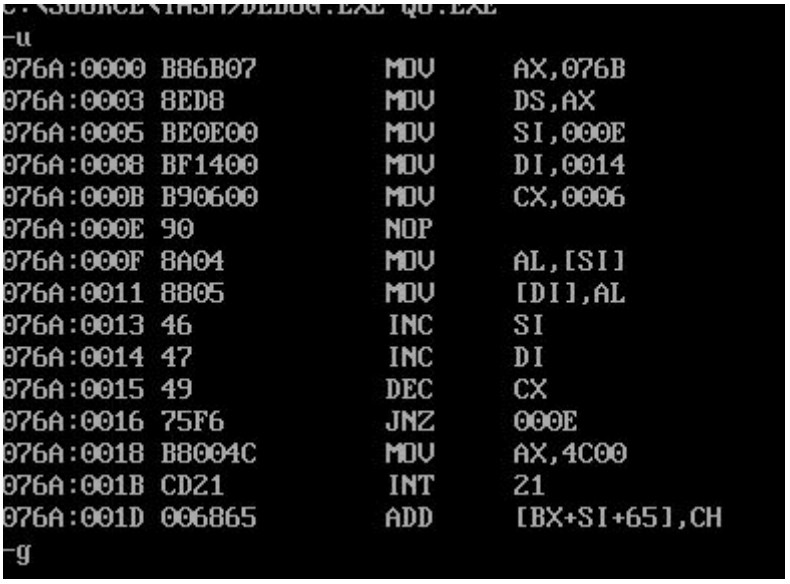
```
  int 21h
end
```

```
AX=076B  BX=0000  CX=0027  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=0027  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 B402          MOV     AH,02
-t

AX=026B  BX=0000  CX=0027  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=026B  BX=6B00  CX=0027  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 BF1300        MOV     DI,0013
-t

AX=026B  BX=6B00  CX=0027  DX=0000  SP=0000  BP=0000  SI=0000  DI=0013
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:0013=31
-
```

```
AX=026B  BX=6B31  CX=0027  DX=0000  SP=0000  BP=0000  SI=0000  DI=0013
DS=076B  ES=075A  SS=0769  CS=076A  IP=000E   NV UP EI PL NZ NA PO NC
076A:000E B81100        MOV     AX,0011
-t

AX=0011  BX=6B31  CX=0027  DX=0000  SP=0000  BP=0000  SI=0000  DI=0013
DS=076B  ES=075A  SS=0769  CS=076A  IP=0011   NV UP EI PL NZ NA PO NC
076A:0011 BE0C00        MOV     SI,000C
-t

AX=0011  BX=6B31  CX=0027  DX=0000  SP=0000  BP=0000  SI=000C  DI=0013
DS=076B  ES=075A  SS=0769  CS=076A  IP=0014   NV UP EI PL NZ NA PO NC
076A:0014 8A5404        MOV     DL,[SI+04]                      DS:0010=45
-t

AX=0011  BX=6B31  CX=0027  DX=0045  SP=0000  BP=0000  SI=000C  DI=0013
DS=076B  ES=075A  SS=0769  CS=076A  IP=0017   NV UP EI PL NZ NA PO NC
076A:0017 B8004C        MOV     AX,4C00
-t

AX=4C00  BX=6B31  CX=0027  DX=0045  SP=0000  BP=0000  SI=000C  DI=0013
DS=076B  ES=075A  SS=0769  CS=076A  IP=001A   NV UP EI PL NZ NA PO NC
076A:001A CD21          INT     21
```

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

```
model small

.8086
.data
  source db "hello$"
  dest db 6 dup(0)

.code
  mov ax,@data
  mov ds,ax
  mov si, offset source
  mov di, offset dest
  mov cx, 0006h
  loop: nop
  mov al, [si]
  mov [di], al
  inc si
  inc di
  dec cx
  jnz loop
  mov ax, 4c00h
  int 21h
end
```



```
Program terminated normally
-d 076b:0000
076B:0000  04 88 05 46 47 49 75 F6-B8 00 4C CD 21 00 68 65   ...FGIu...L.!.he
076B:0010  6C 6C 6F 24 68 65 6C 6C-6F 24 0E 50 34 E8 6C 35   llo$hello$.P4.l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
```

hello copied again

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

```
model small

.8086
.data
  str db "123456$"

.code
  mov ax, @data
  mov ds, ax
  mov si, offset str
  mov cx, 0006h
  mov bl, 05h
  mov di, offset [str + 3]
  up: nop
  inc si
  inc di
  dec bl
  jnz up
  go: nop
  mov al, [si]
  mov [di], al
  dec si
  dec di
  dec cx
  jnz go
  mov ax, 4c00h
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q9.EXE
-u
076A:0000 B86C07        MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 BE0600        MOV     SI,0006
076A:0008 B90600        MOV     CX,0006
076A:000B B305          MOV     BL,05
076A:000D BF0900        MOV     DI,0009
076A:0010 90            NOP
076A:0011 46            INC     SI
076A:0012 47            INC     DI
076A:0013 FECB          DEC     BL
076A:0015 75F9          JNZ     0010
076A:0017 90            NOP
076A:0018 8A04          MOV     AL,[SI]
076A:001A 8805          MOV     [DI],AL
076A:001C 4E            DEC     SI
076A:001D 4F            DEC     DI
076A:001E 49            DEC     CX
076A:001F 75F6          JNZ     0017
-g

Program terminated normally
-
```

```
Program terminated normally
-d 076b:0000
076B:0000  90 46 47 FE CB 75 F9 90-8A 04 88 05 4E 4F 49 75   .FG..u......NOIu
076B:0010  F6 B8 00 4C CD 21 31 32-33 31 32 33 34 35 36 35   ...L.!1231234565
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
```

copied with overlap

10. Program to interchange two blocks of data

```
model small

.8086
.data
  str1 db "hello$"
  str2 db "world$"

.code
  mov ax, @data
  mov ds, ax
  mov si, offset str1
  mov di, offset str2
```

```
  mov cx, 0006h
  loop: nop
  mov al, [si]
  mov bl, [di]
  mov [si], bl
  mov [di], al
  inc si
  inc di
  dec cx
  jnz loop
  mov ax,4c00h
  int 21h
end
```

```
C:\SOURCE\TASM>DEBUG.EXE Q10.EXE
-u
076A:0000 B86C07        MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 BE0200        MOV     SI,0002
076A:0008 BF0800        MOV     DI,0008
076A:000B B90600        MOV     CX,0006
076A:000E 90            NOP
076A:000F 8A04          MOV     AL,[SI]
076A:0011 8A1D          MOV     BL,[DI]
076A:0013 881C          MOV     [SI],BL
076A:0015 8805          MOV     [DI],AL
076A:0017 46            INC     SI
076A:0018 47            INC     DI
076A:0019 49            DEC     CX
076A:001A 75F2          JNZ     000E
076A:001C B8004C        MOV     AX,4C00
076A:001F CD21          INT     21
-g

Program terminated normally
```

```
Program terminated normally
-d 076b:0000
076B:0000  04 8A 1D 88 1C 88 05 46-47 49 75 F2 B8 00 4C CD   .......FGIu...L.
076B:0010  21 00 77 6F 72 6C 64 24-68 65 6C 6C 6F 24 6C 35   !.world$hello$l5
076B:0020  89 1E 5C 04 72 05 83 0E-7A 04 04 BE 42 58 B4 00   ..\.r...z...BX..
076B:0030  8C 0E 50 34 E8 55 35 89-1E 5E 04 72 05 83 0E 7A   ..P4.U5..^.r...z
076B:0040  04 08 83 3E 7A 04 00 75-03 E9 AE 00 53 8B 1E 38   ...>z..u....S..8
076B:0050  07 C4 7F 02 83 C7 18 3B-3F 76 09 E8 2B F8 5B 73   .......;?v..+.[s
076B:0060  07 E8 EC F7 87 7F 02 5B-8B D7 A1 56 04 AB A1 74   .......[...V...t
076B:0070  07 AB A1 84 07 AB A1 94-07 AB A1 AC 07 AB B8 01   ................
```

“hello” and “world” reversed