

## ASM -LAB



NAME- SWARNENDU BANERJEE

ROLL NO.- 002311001016

SECTION : A1

ASSIGNMENT-1

1. Write an Assembly Language Program to add two sixteen-bit numbers. The numbers are stored in DS: 0030H and DS: 0040H. Store the result in DS: 0050H, DS: 0051H, and DS: 0052H.

**Code:**

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

main proc
mov ax, @data
mov ds, ax
mov cl, 00h
mov si, 0030h
mov ax, [si]
mov si, 0040h
mov bx, [si]
add bx, ax
adc cl, cl
mov si, 0050h
mov [si], bx
add si, 02h
mov [si], cl

int 03h
mov ah, 4ch
int 21h
main endp
end main
```

**Output:**

```

C:\>debug q1add.exe
-t

AX=076C BX=0000 CX=0024 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 3D.02  FF.0e
-e 076c:0040
076C:0040 E4.04  40.33
-g=0000

AX=EE02 BX=2106 CX=0001 DX=0000 SP=0100 BP=0000 SI=0052 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=001F  NU UP EI PL NZ NA PO NC
076A:001F CC          INT     3
-d 0050,0052
076C:0050 06 21 01          .!.
```

2. Write an Assembly Language Program to subtract an 8-bit numbers stored in DS: 0030H from a number stored in DS: 0040H using 2's complement method. Store the result in DS: 0050H, and DS: 0051H.

**Code:**

```

.model small
.code

main proc
mov ax, @data
mov ds, ax
mov si, 0030h
mov al, [si]
not al
inc al
mov si, 0040h
add al, [si]
jc L1
not al
inc al
L1: mov si, 0050h
mov [si], al
cmc
mov ah, 00
```

```

adc ah, ah
inc si
mov [si], ah

```

```

int 03h
mov ah, 4ch
int 21h
main endp
end main

```

### Output:

```

C:\>debug q2sub.exe
-t
AX=076C BX=0000 CX=002B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 3D.08
-e 076c:0040
076C:0040 E4.05
-g=0000
AX=0103 BX=0000 CX=002B DX=0000 SP=0000 BP=0000 SI=0051 DI=0000
DS=076C ES=075A SS=0769 CS=076A IP=0026  NU UP EI PL NZ NA PO NC
076A:0026 CC          INT     3
-d 0050,0051
076C:0050 03 01

```

- Write a program to transfer a block of 8 data bytes from memory location DS: 0030H to DS: 0040H.

### Code:

```

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

mov ax, @data
mov ds, ax
mov es, ax
mov si, 0030h

```

```

mov di, 0040h
mov cx, 0008h
cld

```

```

l1:
movsb
loop l1

```

```

mov ah,03h
mov ah, 4ch
int 21h

```

```

main endp
end main

```

### Output:

```

C:\>debug q3block.exe
-t
AX=076B BX=0000 CX=001A DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076b:0030
076B:0030 00.1    52.2    50.3    E8.4    EA.5    48.6    83.7    C4.8

-g=0000

Program terminated normally
-d 076b:0040,0047
076B:0040 01 02 03 04 05 06 07 08

```

4. Write an 8086 Assembly Language Program for the addition of 7 eight-bit numbers stored from DS: 0030H. Store the result in DS: 0050H and DS: 0051H.

### Code:

```

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

mov ax, @data
mov ds, ax

```

```

mov si, 0030h
mov cx, 0007h
mov ax, 0000h
mov bl, 00h

```

```

sum_loop:
add al, [si]
jnc l2
inc bl
l2: inc si
loop sum_loop

```

```

mov si, 0050h
mov [si], al
inc si
mov [si], bl

```

```

int 03h
mov ah, 03h
mov ah, 4ch
int 21h

```

```

main endp
end main

```

### Output:

```

C:\>debug q4add7.exe
-t

AX=076C BX=0000 CX=0028 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 3D.1    FF.2    FF.3    74.4    03.ef    E9.f    ED.a

-g=0000

AX=0012 BX=0001 CX=0000 DX=0000 SP=0100 BP=0000 SI=0051 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0021  NU UP EI PL NZ NA PO NC
076A:0021 CC          INT     3
-d 076c:0050,0051
076C:0050 12 01

```

5. Write an 8086 Assembly Language Program for the addition of 5 sixteen-bit numbers stored from DS: 0030H. Store the result in DS: 0050H, DS: 0051H, DS: 0052H.

**Code:**

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

mov ax, @data
mov ds, ax

mov si, 0030h
mov cx, 0005h
mov ax, 0000h
mov bl, 0000h

sum_loop:
add ax, [si]
jnc l2
inc bl
l2: add si, 0002h
loop sum_loop

mov si, 0050h
mov [si], ax
add si, 0002h
mov [si], bl

int 03h
mov ah, 03h
mov ah, 4ch
int 21h

main endp
end main
```

Output:

```

C:\>debug a1q5.exe
-t

AX=076C BX=0000 CX=002A DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 3D.1      FF.01  FF.2   74.02  03.3   E9.03  ED.4   00.04
076C:0038 C4.5      5E.05

-g=0000

AX=0F0F BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0052 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0025  NU UP EI PL NZ NA PE NC
076A:0025 CC          INT     3
-d 076c:0050,0052
076C:0050 0F 0F 00

```

6. Write an Assembly Language Program for the addition of five BCD numbers stored from DS: 0030H. Store the result in DS: 0040H and DS: 0041H.

Code:

```

.model small
.stack 100h
.data
.code

```

```

main proc
mov ax, @data
mov ds, ax
mov si, 0030h
mov cx, 0005h
mov ax, 0000h
mov bl, 0000h

```

```

l2: add al,[si]
daa
jnc l1
inc bl
l1: inc si
loop l2

```

```

mov si, 0040h
mov [si], al
inc si
mov [si], bl

```



```
int 03h
mov ah, 03h
mov ah, 4ch
int 21h
```

```
main endp
end main
```

Output:

```
C:\>debug a1q6.exe
-t

AX=076C BX=0000 CX=0029 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NV UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 3D.11  FF.22  FF.33  74.44  03.55

-g=0000

AX=0065 BX=0001 CX=0000 DX=0000 SP=0100 BP=0000 SI=0041 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0022  NV UP EI PL NZ NA PE NC
076A:0022 CC          INT     3
-d 076c:0040,0041
076C:0040 65 01                                     e.
```

7. Write an Assembly Language Program to subtract a BCD number stored in DS: 0040H from a BCD number stored in DS: 0050H. Store the result in DS: 0060H and DS: 0061H.

**Code:**

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

main proc
mov ax, @data
mov ds, ax

mov si, 0050h
mov al, [si]
mov si, 0040h

mov ah, 00h
```

```

sub al, [si]
das
jnc l2
adc ah, 00h

```

```

l2: mov si, 0060h
mov [si], ax

```

```

int 03h
mov ah, 4ch
int 21h

```

```

main endp
end main

```

Output:

```

C:\>debug a1q7.exe
-t

AX=076C BX=0000 CX=0021 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 E4.09

-e 076c:0050
076C:0050 C4.14

-g=0000

AX=0005 BX=0000 CX=0021 DX=0000 SP=0100 BP=0000 SI=0060 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=001C  NU UP EI PL NZ AC PE NC
076A:001C CC          INT     3
-d 076c:0060,0061
076C:0060 05 00          . .

```

8. Write an Assembly Language Program to multiply two eight bit number stored in DS: 0040H and DS: 0050H. Store the result from DS: 0060H.

**Code:**

```

.model small
.stack 100h
.data
.code

```

```

main proc

```

```
mov ax, @data
mov ds, ax
```

```
mov si, 0040h
mov al, [si]
mov si, 0050h
mov bl, [si]
```

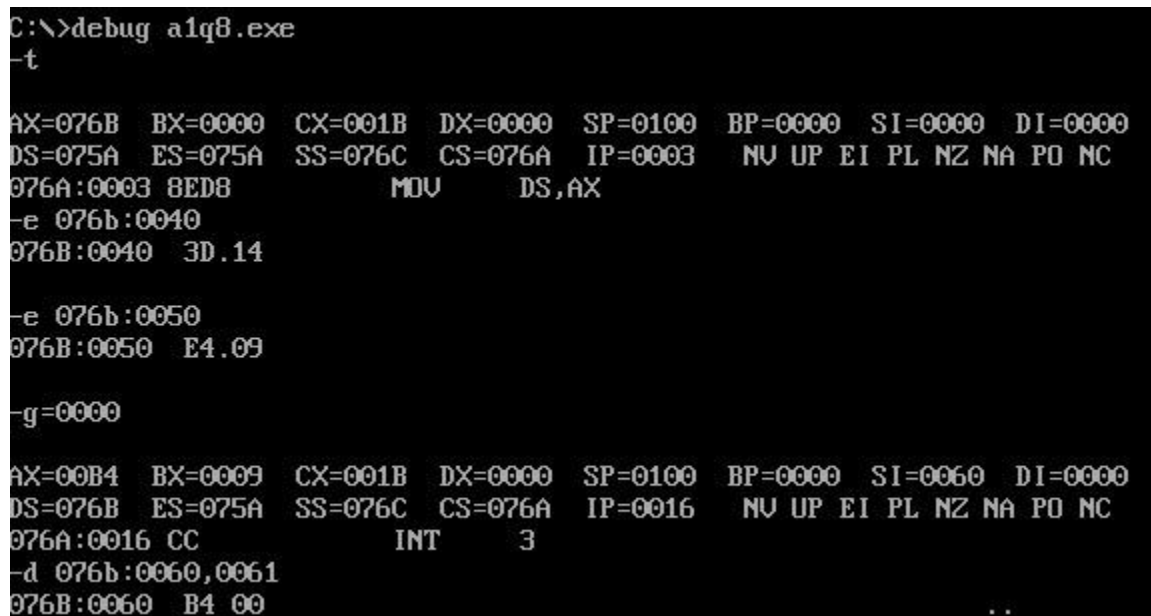
```
mul bl
```

```
mov si, 0060h
mov [si], ax
```

```
int 03h
mov ah, 4ch
int 21h
```

```
main endp
end main
```

Output:



```
C:\>debug a1q8.exe
-t

AX=076B BX=0000 CX=001B DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076b:0040
076B:0040 3D.14

-e 076b:0050
076B:0050 E4.09

-g=0000

AX=00B4 BX=0009 CX=001B DX=0000 SP=0100 BP=0000 SI=0060 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=0016  NU UP EI PL NZ NA PO NC
076A:0016 CC          INT     3
-d 076b:0060,0061
076B:0060 B4 00
```

9. Write an Assembly Language Program to multiply two sixteen bit number stored in DS:0040H and DS:0050H. Store the result from DS: 0060H.

**Code:**

```
.model small
```

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0040h

mov ax, [si]

mov si, 0050h

mov bx, [si]

mul bx

mov si, 0060h

mov [si], ax

add si, 0002h

mov [si], dx

int 03h

mov ah, 4ch

int 21h

main endp

end main

Output:

```

C:\>debug a1q9.exe
-t

AX=076C BX=0000 CX=0020 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 E4.34  40.12

-e 076c:0050
076C:0050 C4.21  04.43

-g=0000

AX=F4B4 BX=4321 CX=0020 DX=04C5 SP=0100 BP=0000 SI=0062 DI=0000
DS=076C ES=075A SS=076C CS=076A IP=001B  NU UP EI PL NZ NA PO NC
076A:001B CC          INT     3
-d 076c:0060,0063
076C:0060 B4 F4 C5 04          ....

```

10. Write an Assembly Language Program to divide 88H by 33H. Store the quotient in DS: 0060H and remainder in DS: 0061H.

**Code:**

```

.model small
.stack 100h
.data
.code

```

```

main proc
mov ax, @data
mov ds, ax
mov ax, 0088h
mov bl, 33h
div bl
mov si, 0060h
mov [si],al
inc si
mov [si], ah

```

```

int 03h
mov ah, 4ch
int 21h

```

```

main endp
end main

```

Output:

```
C:\>debug a1q10.exe
-t

AX=076B BX=0000 CX=0019 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NV UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-g=0000

AX=2202 BX=0033 CX=0019 DX=0000 SP=0100 BP=0000 SI=0061 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=0014  NV UP EI PL NZ NA PO NC
076A:0014 CC          INT     3
-d 076b:0060,0061
076B:0060 02 22          ."
.
```

11. Write an Assembly Language Program to divide 2222H by 55H. Store the quotient from DS: 0060H and remainder in DS: 0062H.

**Code:**

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

main proc

```
mov ax, @data
mov ds, ax
mov ax, 2222h
mov bx, 0055h
div bx
```

```
mov si, 0060h
mov [si], ax
add si, 2
mov [si], dx
```

```
int 03h
mov ah, 4ch
int 21h
```

```
main endp
end main
```

Output:

```
C:\>debug a1q11.exe
```

```
-t
```

```
AX=076B  BX=0000  CX=001C  DX=0000  SP=0100  BP=0000  SI=0000  DI=0000  
DS=075A  ES=075A  SS=076C  CS=076A  IP=0003  NU UP EI PL NZ NA PO NC  
076A:0003 8ED8          MOV     DS,AX  
-g=0000
```

```
AX=0066  BX=0055  CX=001C  DX=0044  SP=0100  BP=0000  SI=0062  DI=0000  
DS=076B  ES=075A  SS=076C  CS=076A  IP=0017  NU UP EI PL NZ NA PO NC  
076A:0017 CC          INT     3  
-d 076b:0060,0062  
076B:0060 66 00 44          f.D
```