Name : Dripta Patra

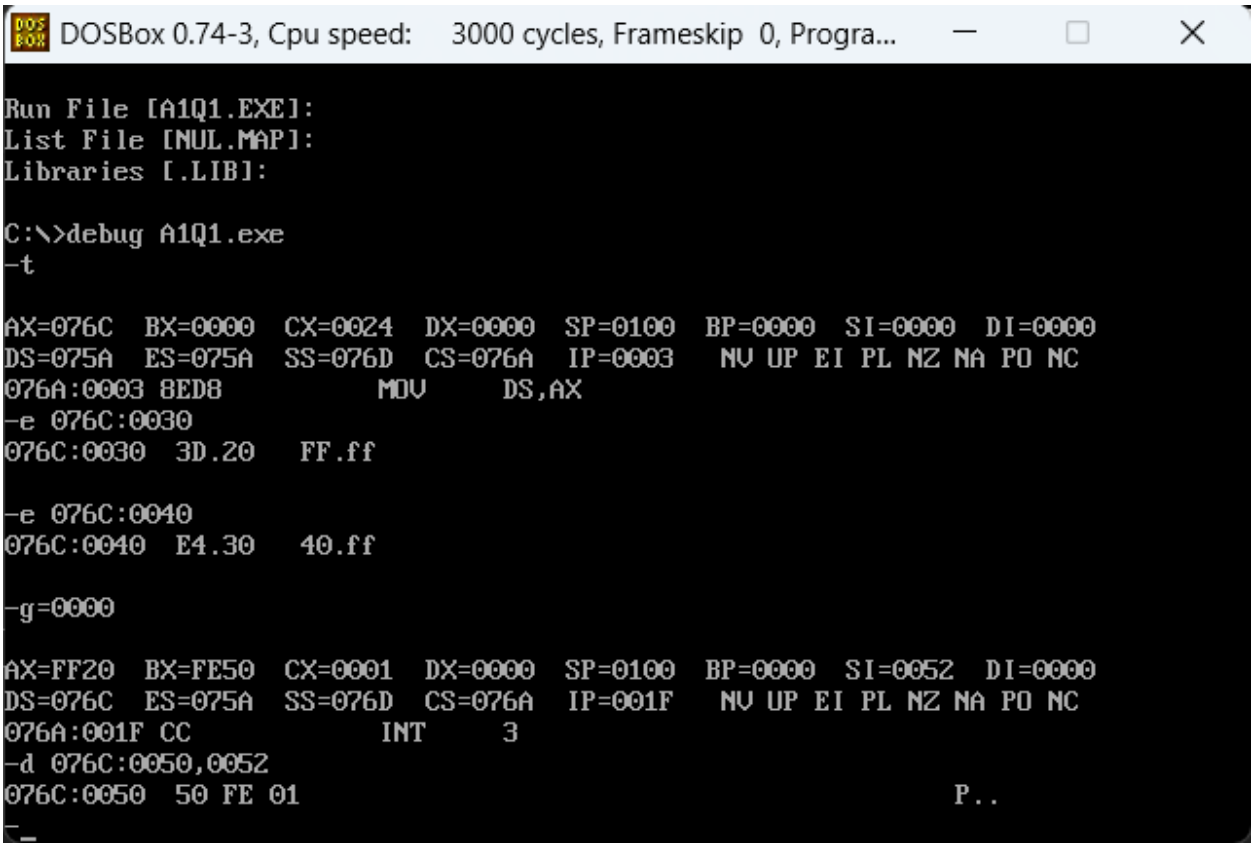
Roll No : 002311001006

Section : A1

Dept : IT UG-2

Q1) 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.

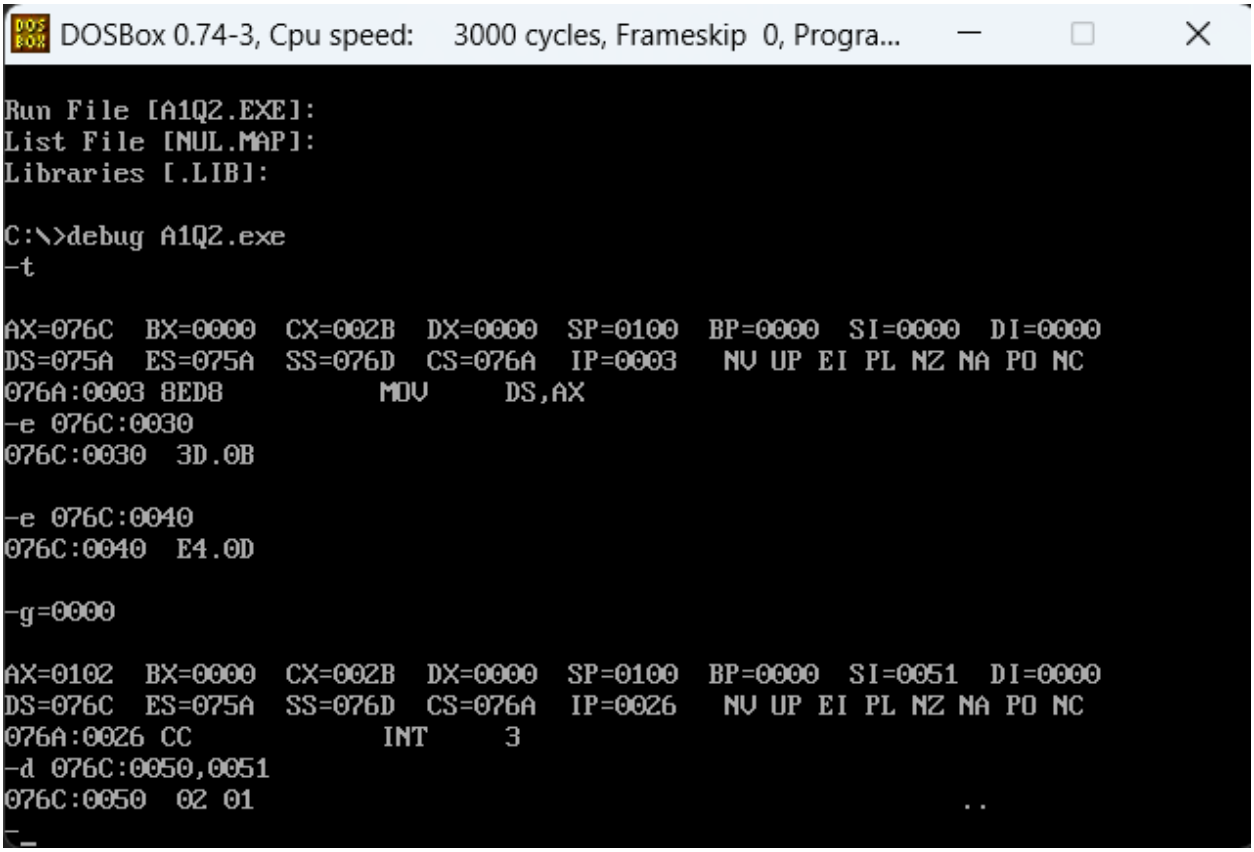
```
.model small
.stack 100h
.data
.code
main proc
        mov ax, @data
        mov ds, ax
        mov cl, 00h
        mov si, 0030h
        mov ax, [si]
        mov si, 0040h
        mov bx, [si]
        add bx, ax
        adc cl, cl
        mov si, 0050h
        mov [si], bx
        add si, 02h
        mov [si],cl
        int 03h
        mov ah, 4ch
        int 21h
main endp
end main
```



Q2) 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.

```
.model small
.stack 100h
.data
.code
main proc
        mov ax, @data
        mov ds, ax
        mov si, 0040h
        mov al, [si]
        not al
        inc al
        mov si, 0030h
        add al, [si]
        jc l1
        not al
        inc al
        l1 :
```

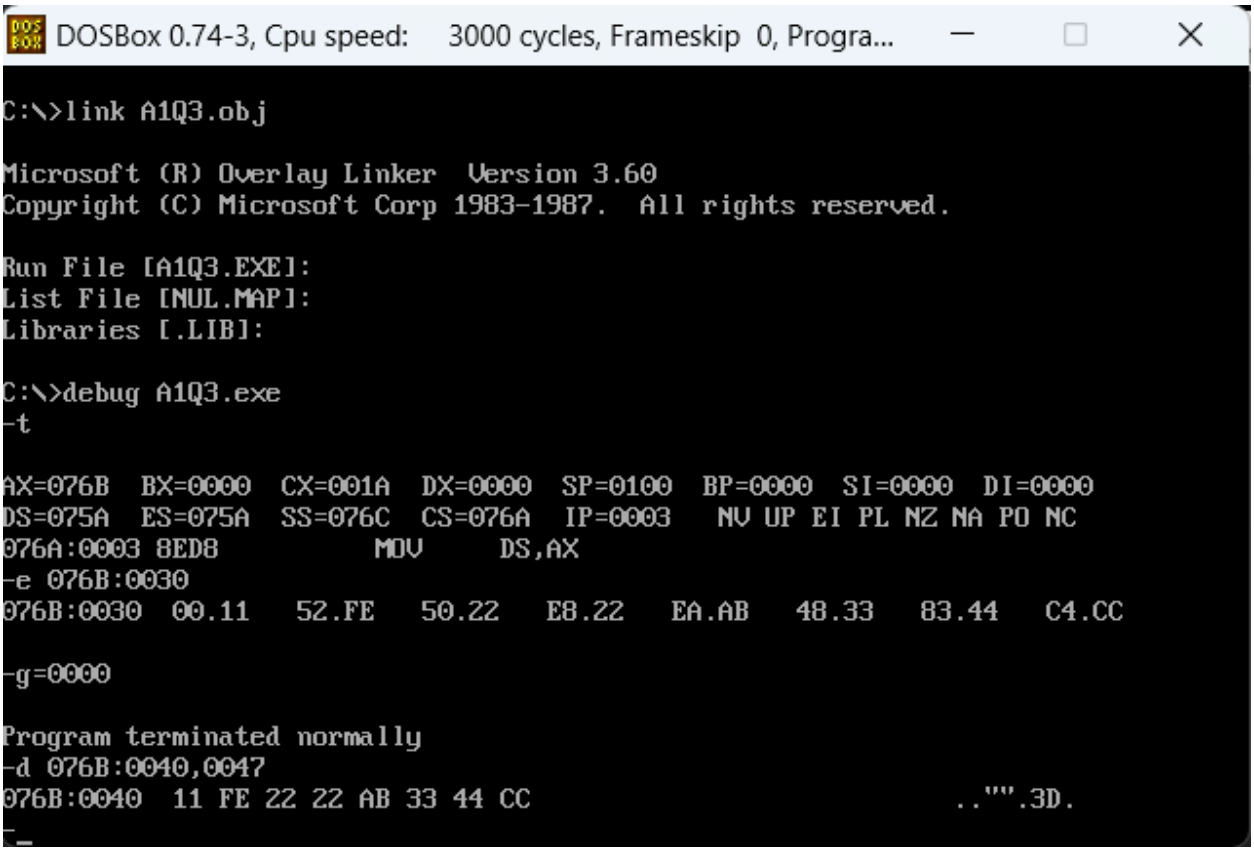
```
        mov si, 0050h
        mov [si], al
        cmc
        mov ah, 00
        adc ah, ah
        inc si
        mov [si], ah
        int 03h
        mov ah, 4ch
        int 21h
main endp
end main
```



Q3) Write a program to transfer a block of 8 data bytes from memory location DS: 0030H to DS: 0040H.

```
.model small
.stack 100h
.data
.code
main proc
        mov ax, @data
```

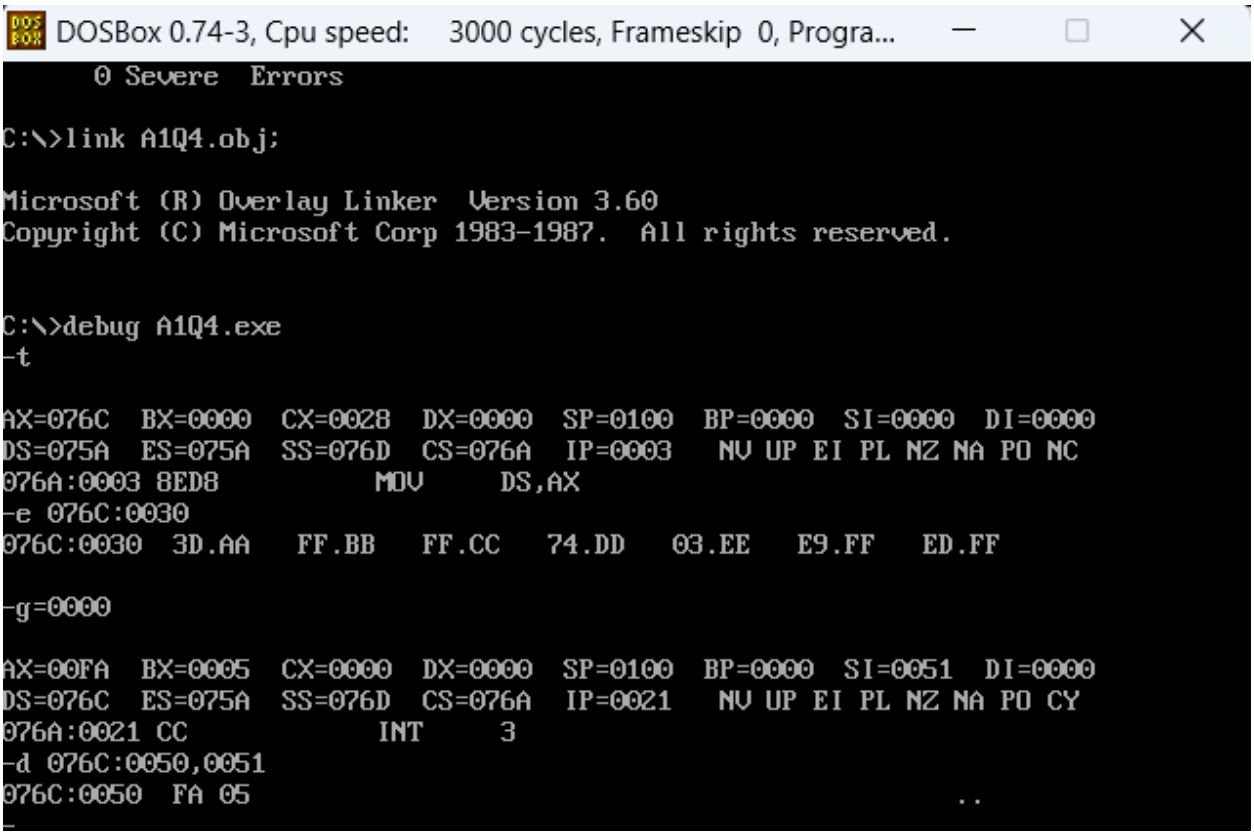
```
        mov ds, ax
        mov es, ax
        mov si, 0030h
        mov di, 0040h
        mov cx, 0008h
        cld
        l1:
        movsb
        loop l1
        mov ah,03h
        mov ah, 4ch
        int 21h
main endp
end main
```



Q4) 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.

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

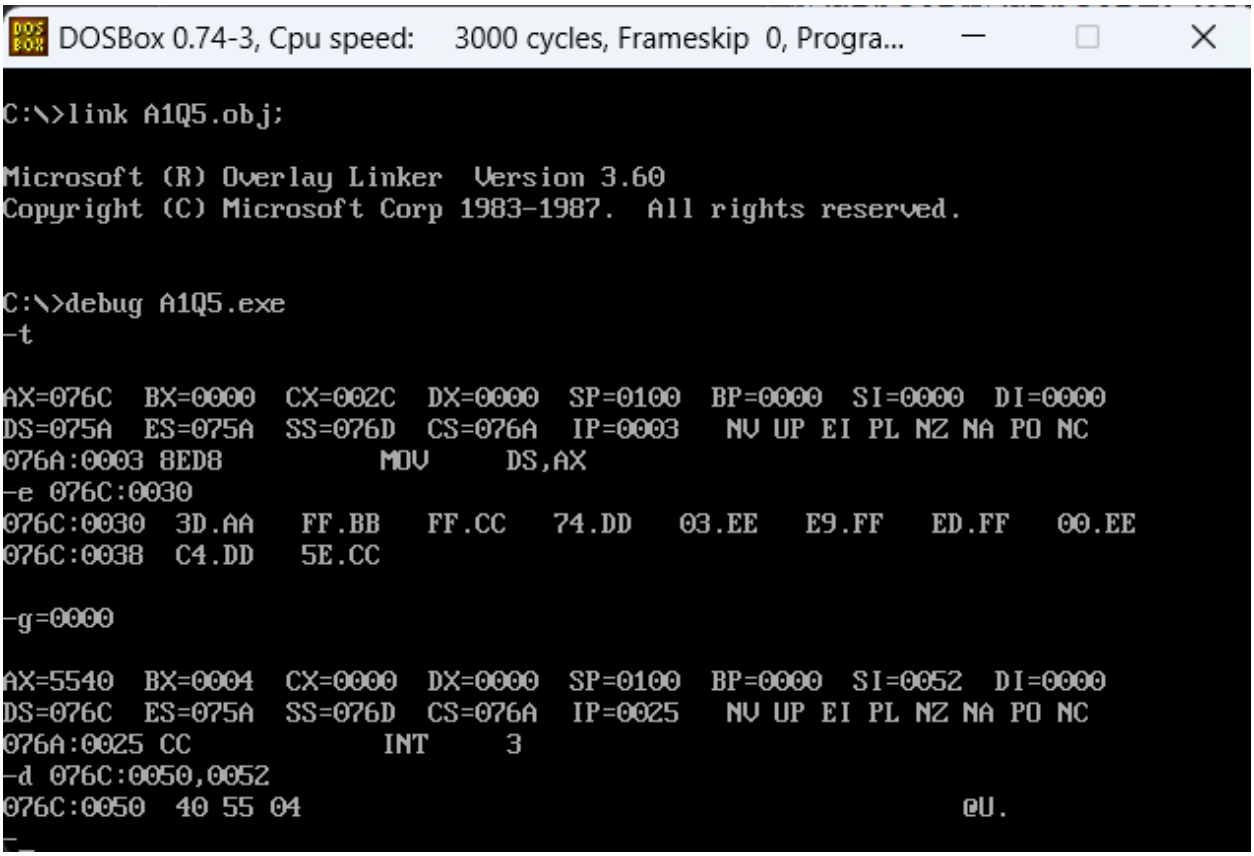
```
.code
main proc
        mov ax, @data
        mov ds, ax
        mov si, 0030h
        mov cx, 0007h
        mov ax, 0000h
        mov bl, 00h
        l1:
        add al, [si]
        jnc l2
        inc bl
        l2 : inc si
        loop l1

        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
```



Q5) 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.

```
.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, 00h
        l1:
        add ax, [si]
        jnc l2
        inc bl
        l2 : add si,02h
        loop l1
```

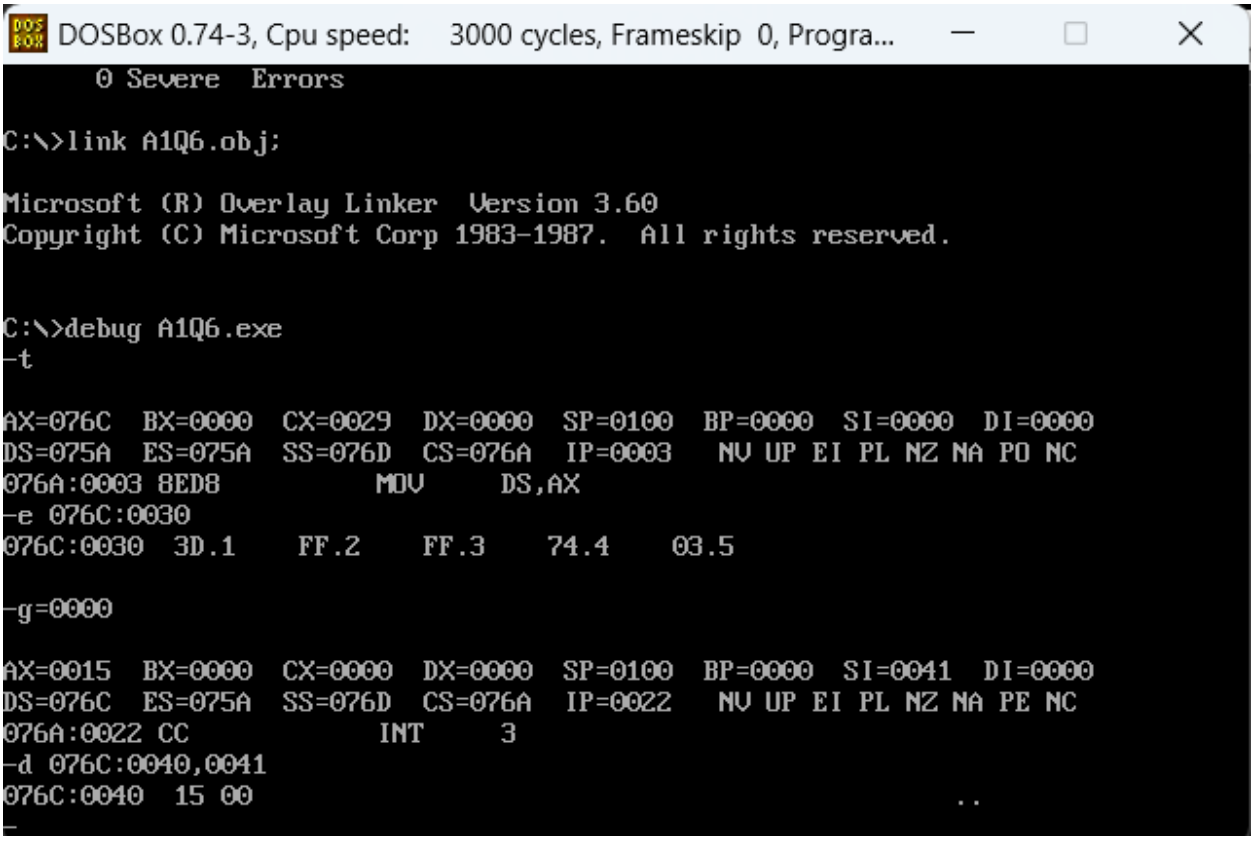
```
        mov si, 0050h
        mov [si], ax
        add si, 02h
        mov [si], bl
        int 03h
        mov ah,03h
        mov ah, 4ch
        int 21h
main endp
end main
```



Q6) 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.

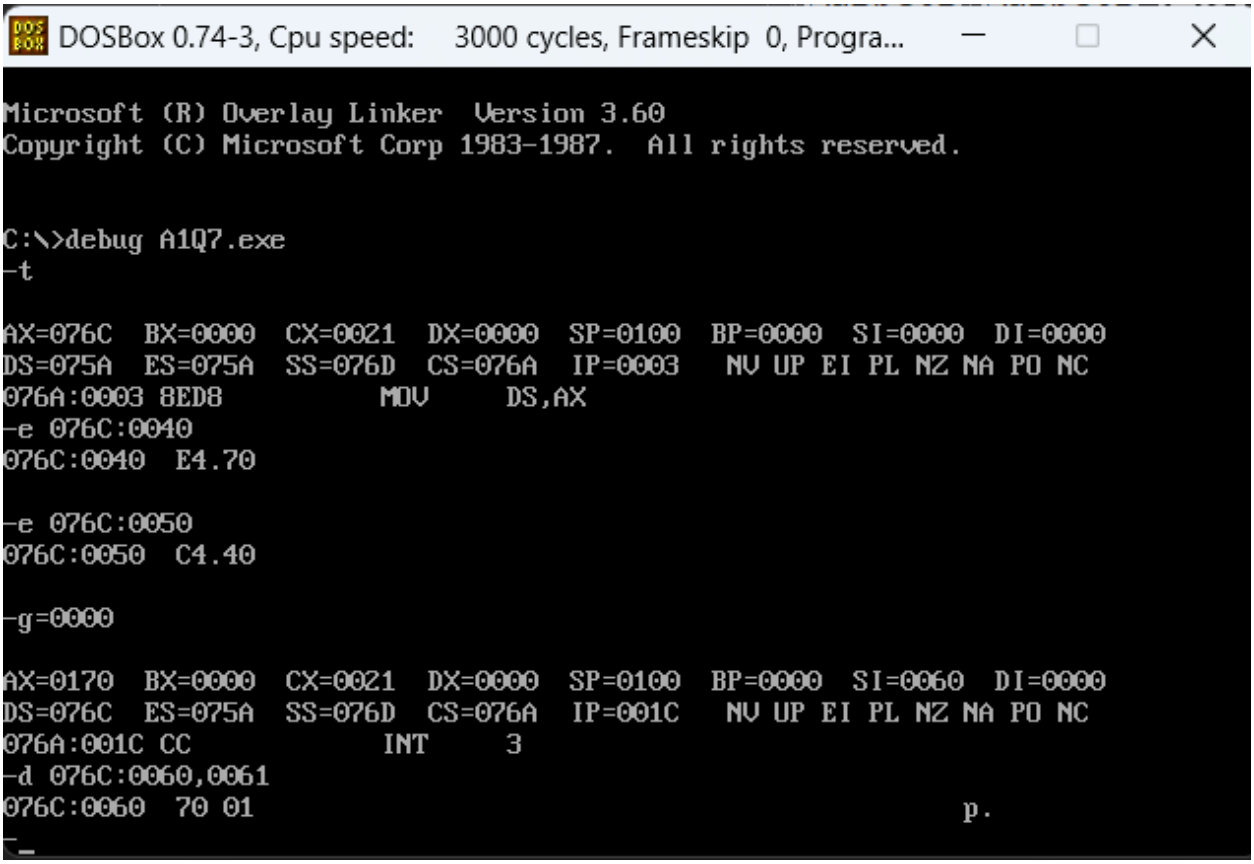
```
.model small
.stack 100h
.data
.code

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
```



Q7) 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.

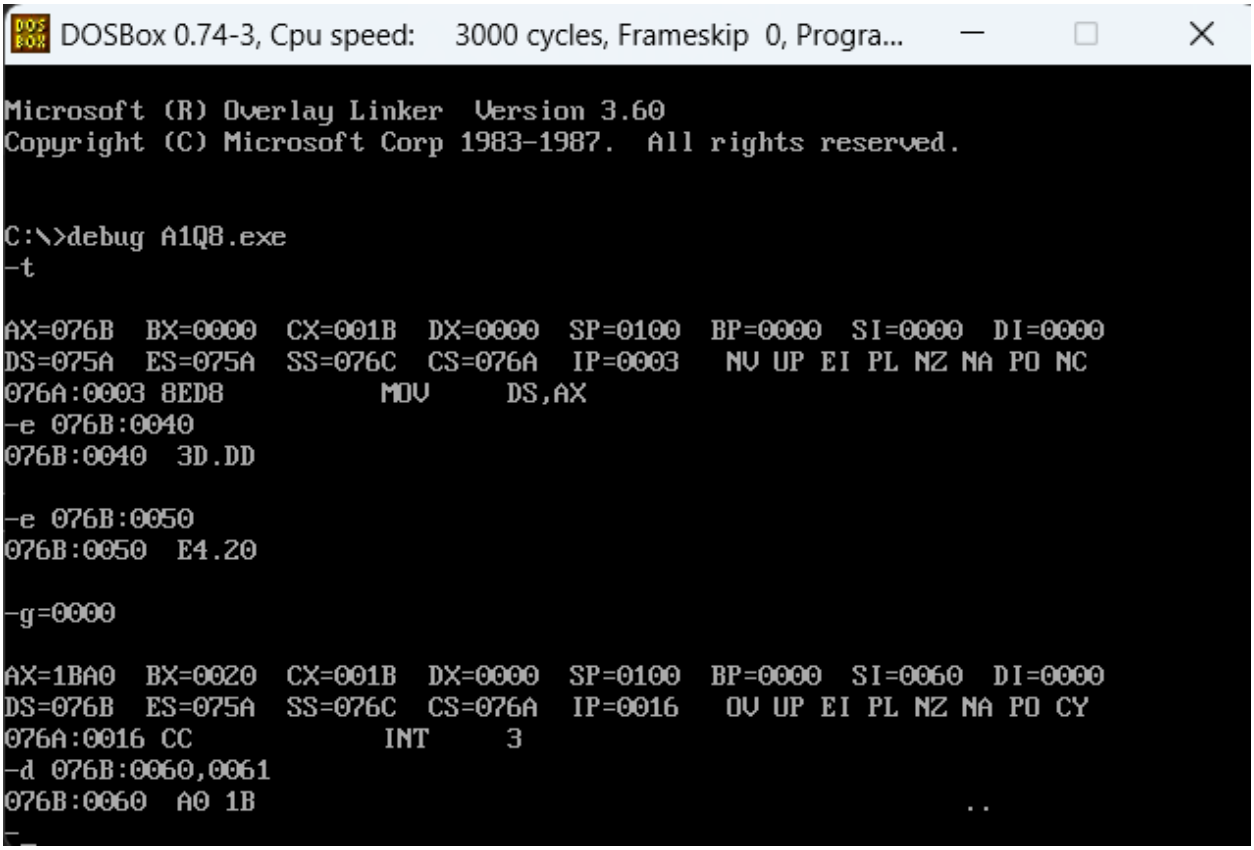
```
.model small
.stack 100h
.data
.code

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
```



Q8) Write an Assembly Language Program to multiply two eight bit number stored in DS: 0040H and DS: 0050H. Store the result from DS: 0060H.

```
.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
```

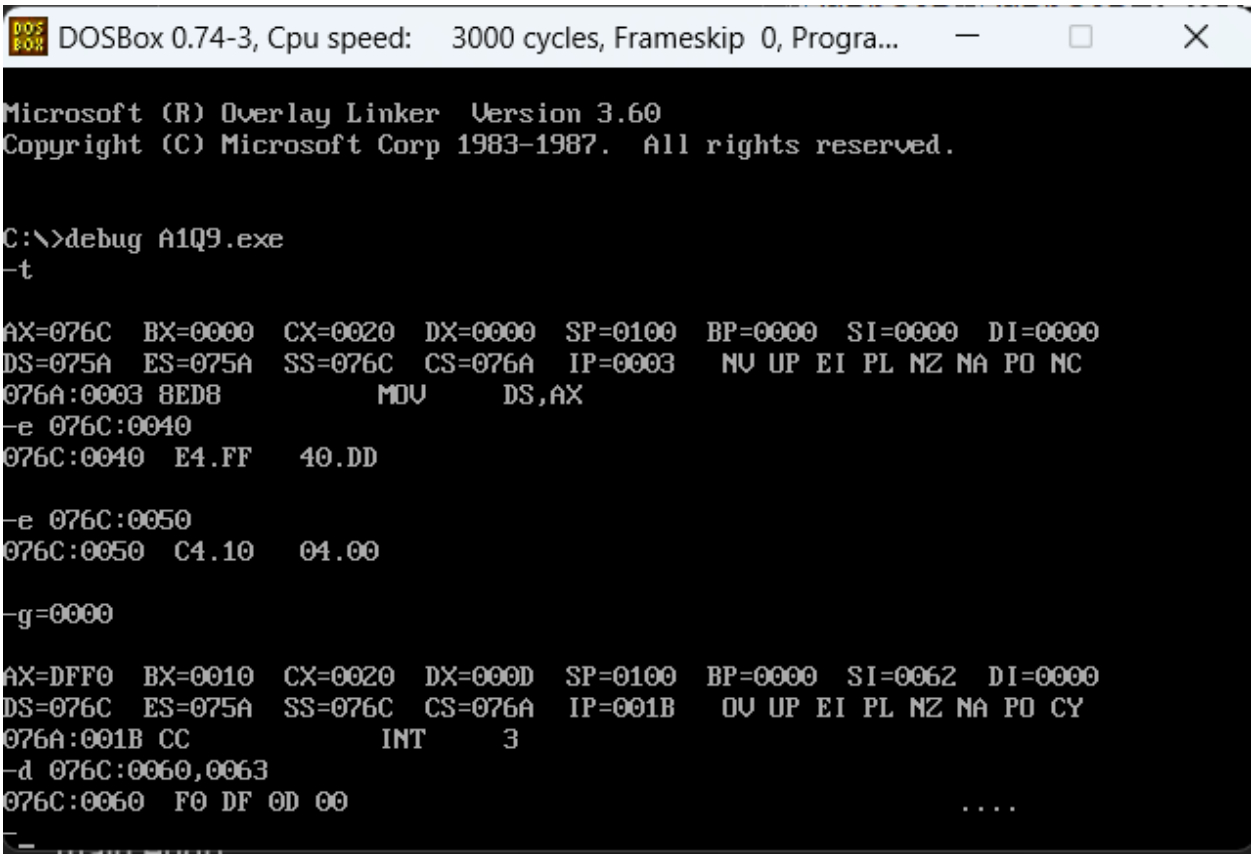
```
        int 21h
main endp
end main
```



Q9) Write an Assembly Language Program to multiply two sixteen bit number stored in DS:0040H and DS:0050H. Store the result from DS: 0060H.

```
.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
```

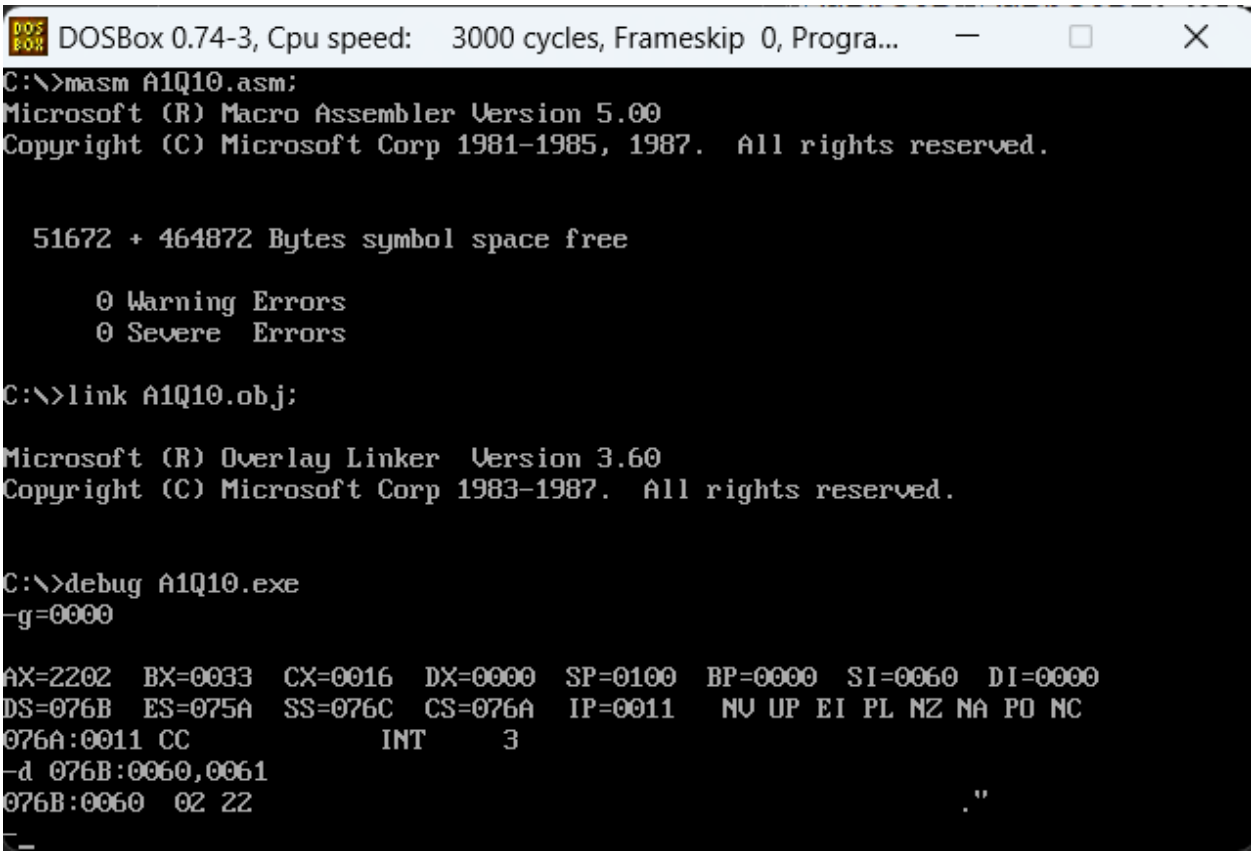
```
        mov [si],ax
        mov si, 0062h
        mov [si],dx
        int 03h
        mov ah, 4ch
        int 21h
main endp
end main
```



Q10) Write an Assembly Language Program to divide 88H by 33H. Store the quotient in DS: 0060H and remainder in DS: 0061H.

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



Q11)Write an Assembly Language Program to divide 2222H by 55H. Store the quotient from DS: 0060H and remainder in DS: 0062H.

```
.model small
.stack 100h
.data
.code
main proc
        mov ax, @data
        mov ds, ax
        mov dx, 0000
        mov ax, 2222h
```

```
        mov bx, 0055h
        div bx
        mov si, 0060h
        mov [si], ax
        mov si, 0062h
        mov [si], dx
        int 03h
        mov ah, 4ch
        int 21h
main endp
end main
```