Name: DEBODIT RAY

Roll Number: 002311001013

Section: A1

Dept: INFORMATION TECHNOLOGY

Year: UG2 Sem 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.

. model small

. stack 100h

. data

. code

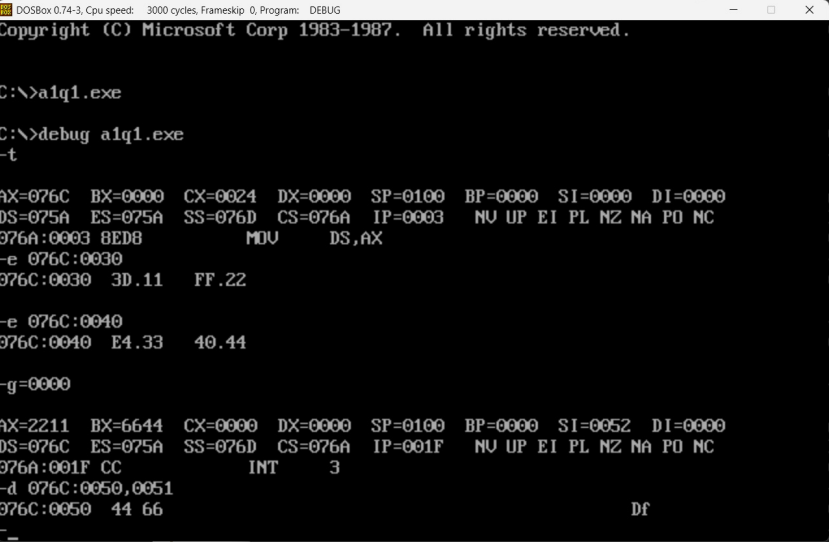
main proc

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

1. 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, 0030h

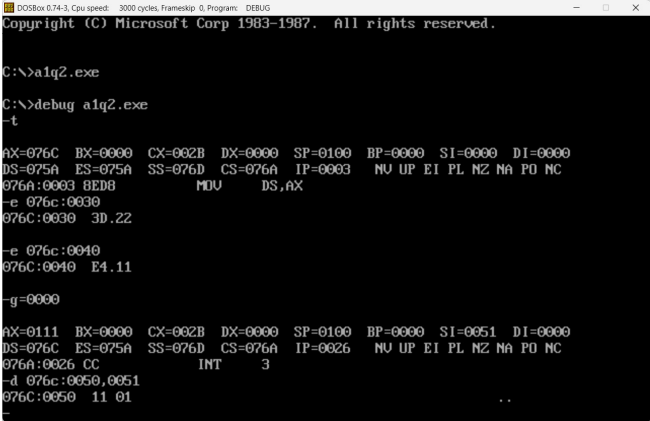
mov al, [si]

not al

add al, 01h

mov si, 0040h

add al, [si]



jc l1

not al

inc al

l1:

mov si, 0050h

mov [si], al

mov ah,00h

cmc

adc ah, ah

inc si

mov [si], ah

int 03h

mov ah,4ch

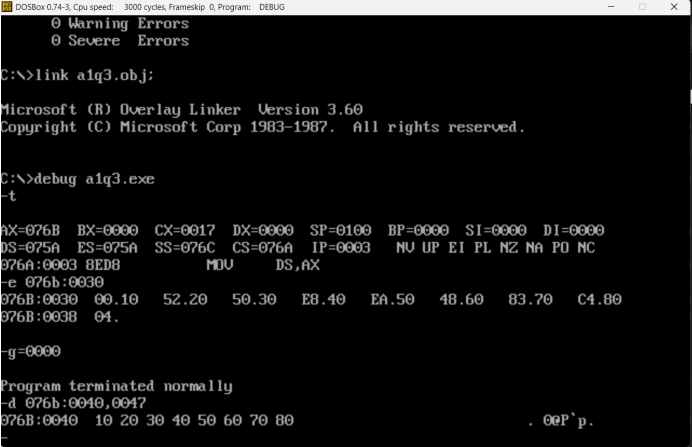
int 21h

main endp

end main

1. 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

cld

mov cx, 0008h

rep movsb

mov ah,4ch

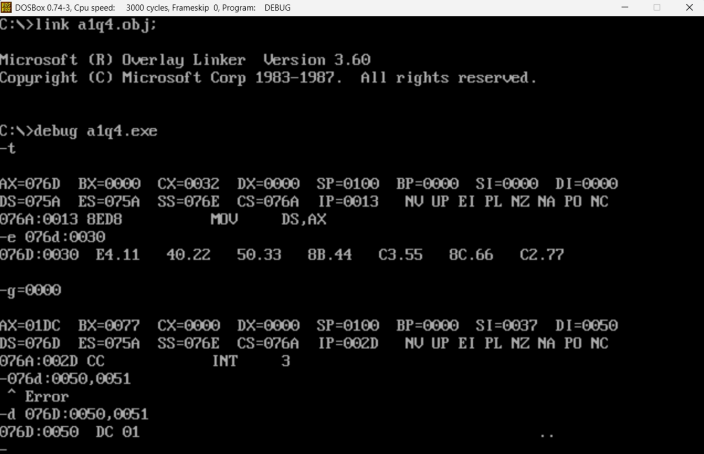
int 21h

main endp

end main

1. 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.

dosseg

. model small

. stack 100h

. data

. code

main proc

mov ax, @data

mov ds, ax

mov si,0030h

mov di, 0050h

mov cx,0007h

mov ax,0000h

mov bl,0000h

l1: mov bl, [si]

add ax, bx

inc si

loop l1

mov [di], ax

int 03h

mov ah,4ch

int 21h

main endp

end main

[Note: 11+22+33+44+55+66+77 = 01DC]

1. 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.

dosseg

. model small

. stack 100h

. data

. code

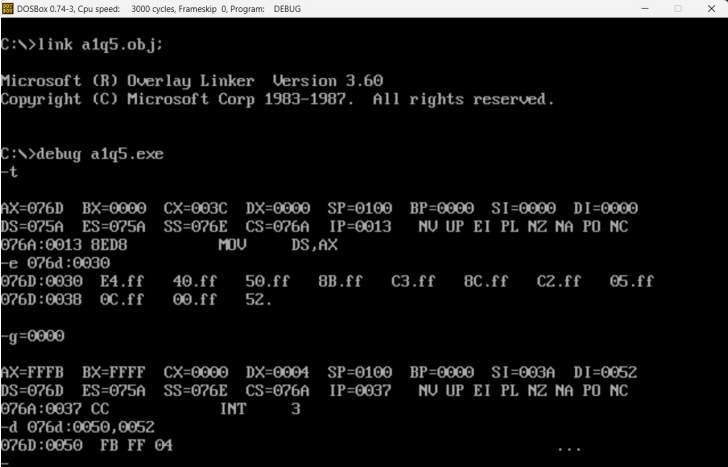
main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov di, 0050h

mov cx, 0005h

mov ax, 0000h

mov bx, 0000h

l1: mov bx, [si]

add ax, bx

adc dl,00h

inc si

inc si

loop l1

mov [di], ax

inc di

inc di

mov [di], dl

int 03h

mov ah,4ch

int 21h

main endp

end main

[Note: FFFF + FFFF + FFFF + FFFF + FFFF = 04FFFB]

1. 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.

dosseg

. model small

. stack 100h

. data

. code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov di, 0040h

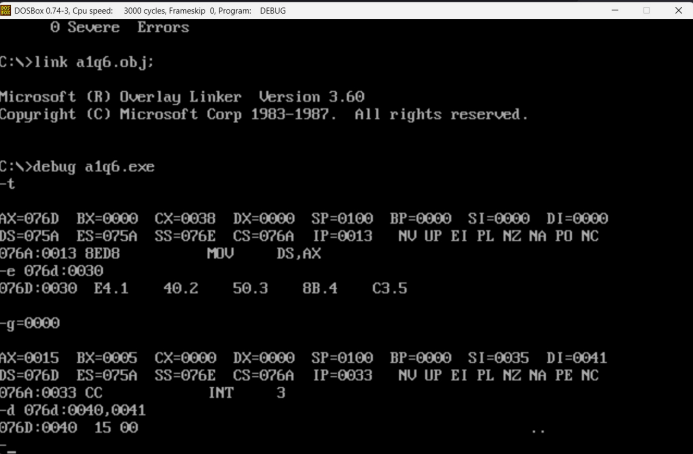
mov cx, 0005h

mov ax, 0000h

mov dl, 00h

l1:

mov bl, [si]

 add al, bl

daa

adc dl, 00h

inc si

loop l1

mov [di], al

inc di

mov [di], dl

int 03h

mov ah, 4ch

int 21h

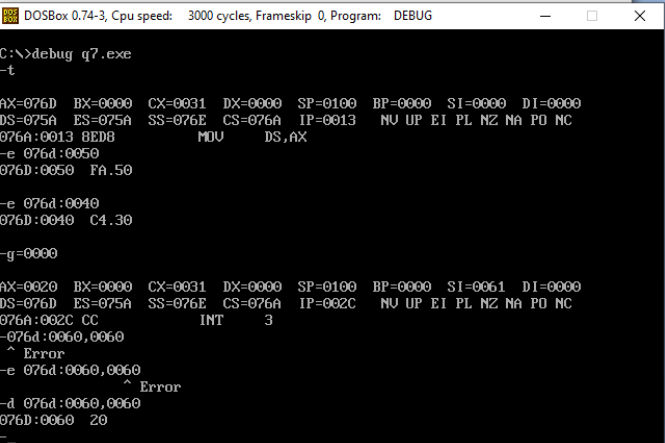
main endp

end main

1. 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.

dosseg

. model small

. stack 100h

. data

. code

main proc

mov ax, @data

mov ds, ax

mov si,0050h

mov al, [si]

mov si,0040h

sub al, [si]

das

mov si,0060h

mov [si], al

mov ah,00h

adc ah, ah

inc si

mov [si], ah

int 03h

mov ah,4ch

int 21h

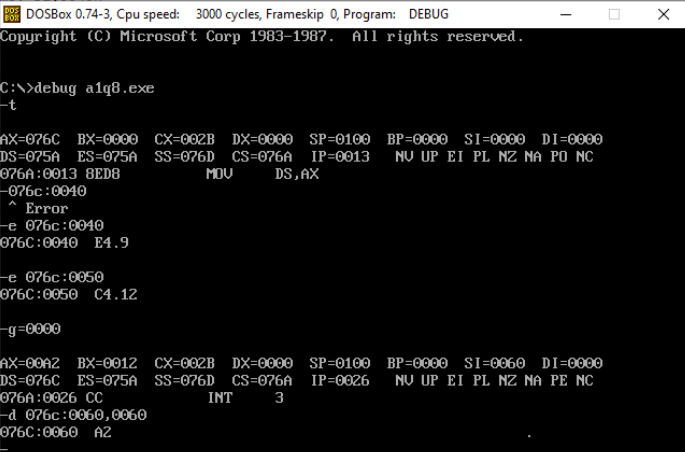
main endp

end main

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

dosseg

. 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

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

dosseg

. model small

. stack 100h

. data

. code

main proc

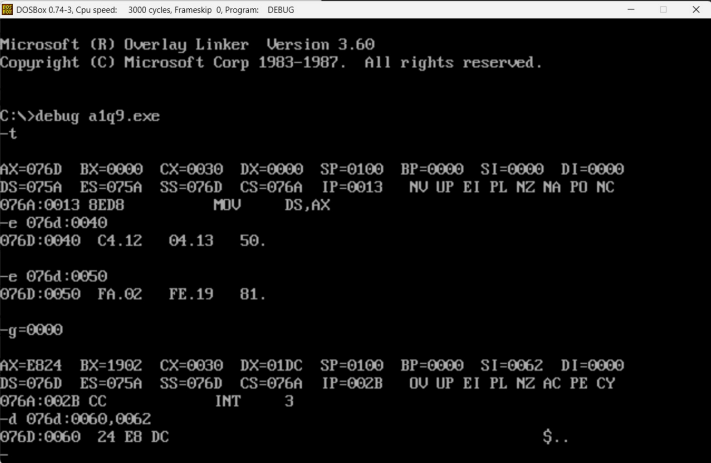
mov ax, @data

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

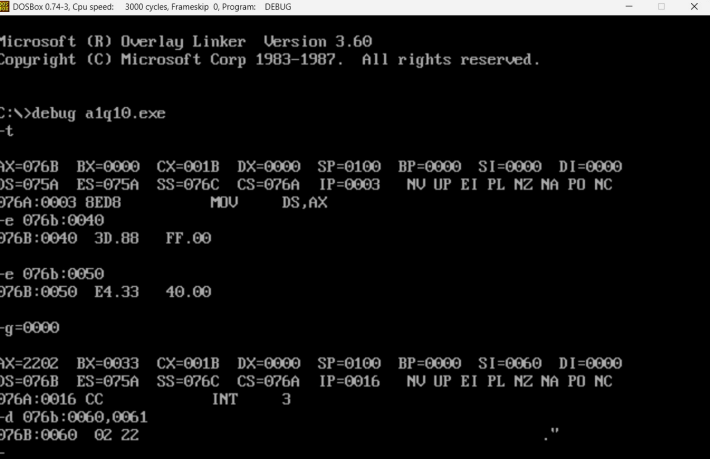
1. Write an Assembly Language Program to divide 88H by 33H. Store the quotient in DS: 0060H and remainder in DS: 0061H.

. model small

. stack 100h

. data

. code



main proc

mov ax, @data

mov ds, ax

mov si,0040h

mov ax, [si]

mov si,0050h

mov bl, [si]

div bl

mov si,0060h

mov [si], ax

int 03h

mov ah,4ch

int 21h

main endp

end main

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

dosseg

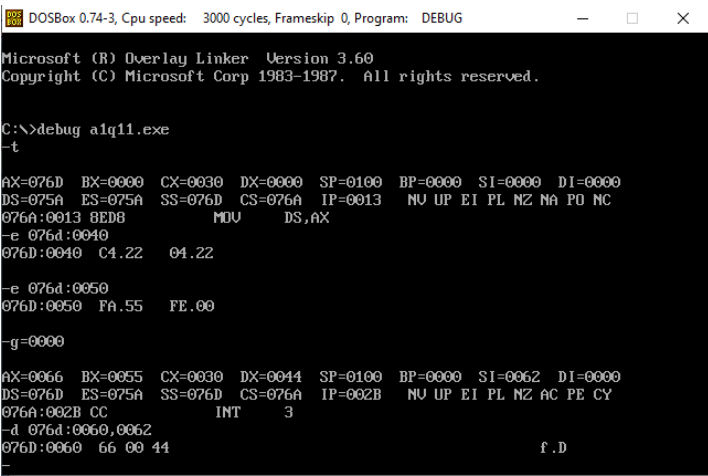
. model small

. stack 100h

. data

. code

main proc



mov ax, @data

mov ds, ax

mov si,0040h

mov ax, [si]

mov si,0050h

mov bx, [si]

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

1. Write an Assembly Language Program to count the number of occurrence of 55H in a string of eight data bytes. The starting address of string is DS: 0030H. Store the count value in DS:0040H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov al, 55h

mov cx, 0008h

mov di, 0030h

mov bl, 00h

l1:

scasb

jnz l2

inc bl

l2:

loop l1

mov si, 0040h

mov [si], bl

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to find out the location where 55H is placed in a string of eight data bytes. The starting address of string is DS: 0030H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov di, 0030h

mov al, 55h

mov cx, 0008h

mov si, 0040h

cld

l1:

scasb

jnz l2

dec di

mov [si], di

add si, 0002h

inc di

l2:

loop l1

int 03h

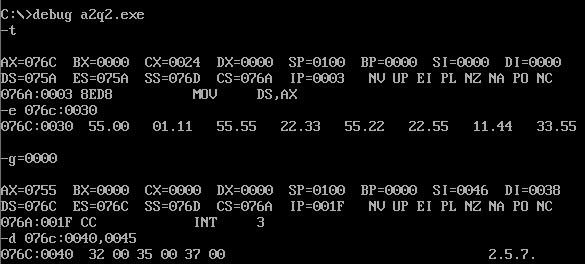
mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to compare two strings. The first string is stored from memory location DS: 0030H and the second sting is stored from DS: 0040H. Consider that the first byte of both strings contain the number of bytes contained in that string. If both strings are found equal, then show a value FFFFH in address DS: 0050H, otherwise show 1111H.

**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 cl, [si]

mov ch, 00h

cld

l1:

cmpsb

jnz l2

loop l1

mov ax, 0ffffh

jmp l3

l2:

mov ax, 01111h

l3:

mov bx, 0050h

mov [bx], ax

int 03h

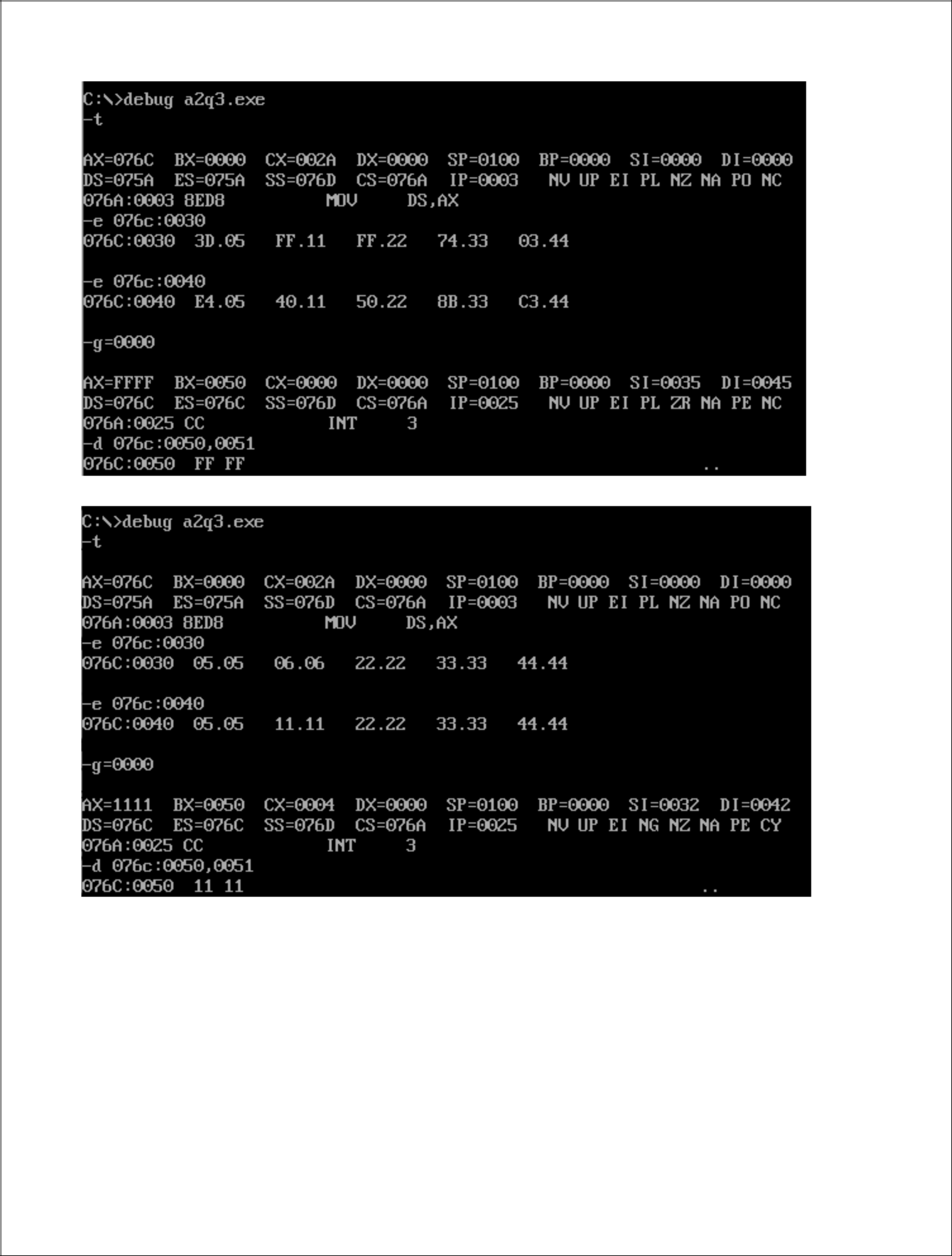
mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to check if a string of five data bytes is palindrome or not. The string is stored from memory location DS: 0030H. If the string is found to be palindrome then place FFFFH in addresses DS: 0040H otherwise place 1111H.

Code:

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov ax, 0005h

mov si, 0030h

mov di, 0030h

add di, ax

dec di

mov bl, 02h

div bl

mov cl, al

mov ch, 00h

l1:

mov al, [si]

mov bl, [di]

cmp al, bl

jnz l2

loop l1

mov ax, 0ffffh

jmp l3

l2:

mov ax, 01111h

l3:

mov bx, 0040h

mov [bx], ax

int 03h

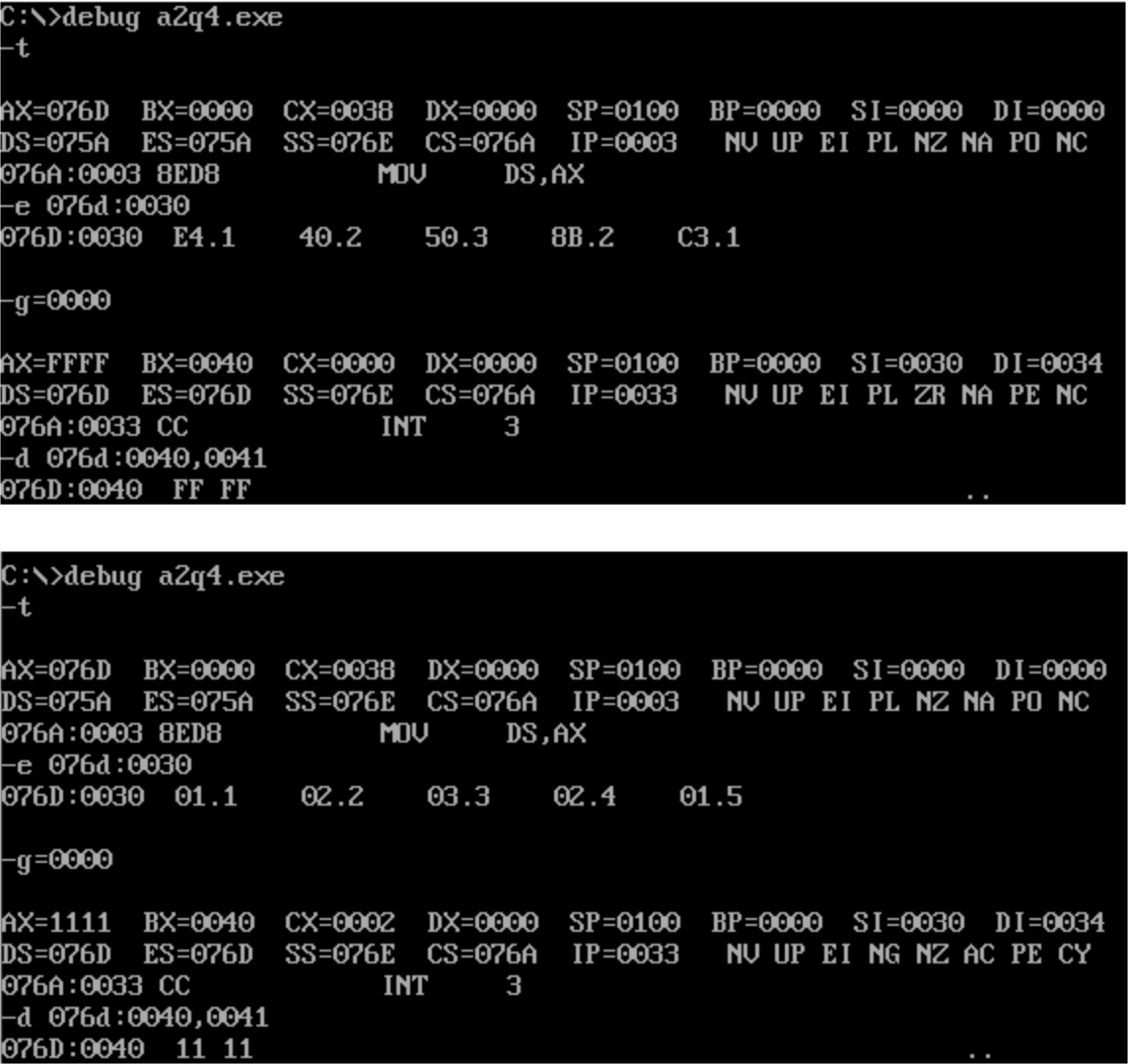
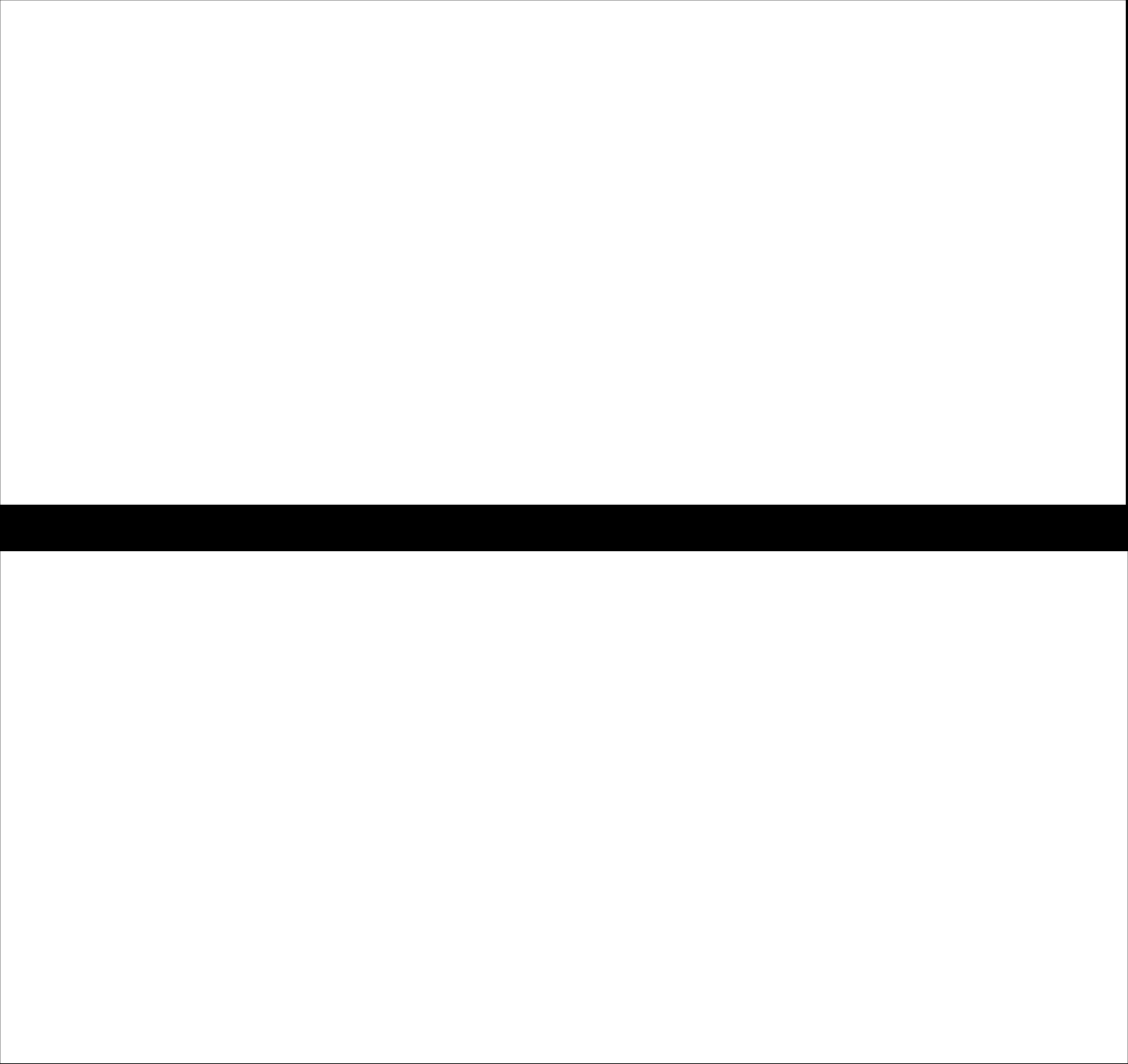
mov ah, 4ch

int 21h

main endp

end main

Output:



1. Write an Assembly Language Program to count the number of positive and negative numbers present in a series of eight data bytes. The starting address of the series is DS: 0040H. Store the count value of positive number in DS: 0040H and count value of negative number in DS: 0041H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov bx, 0000h ; storing +ve in bh, -ve in bl

mov si, 0040h

mov cx, 0008h

l1: mov al, [si]

rol al, 01h

inc si

jc l2

inc bh

jmp l3

l2: inc bl

l3: loop l1

mov si, 0040h

mov [si], bh

inc si

mov [si], bl

int 03h

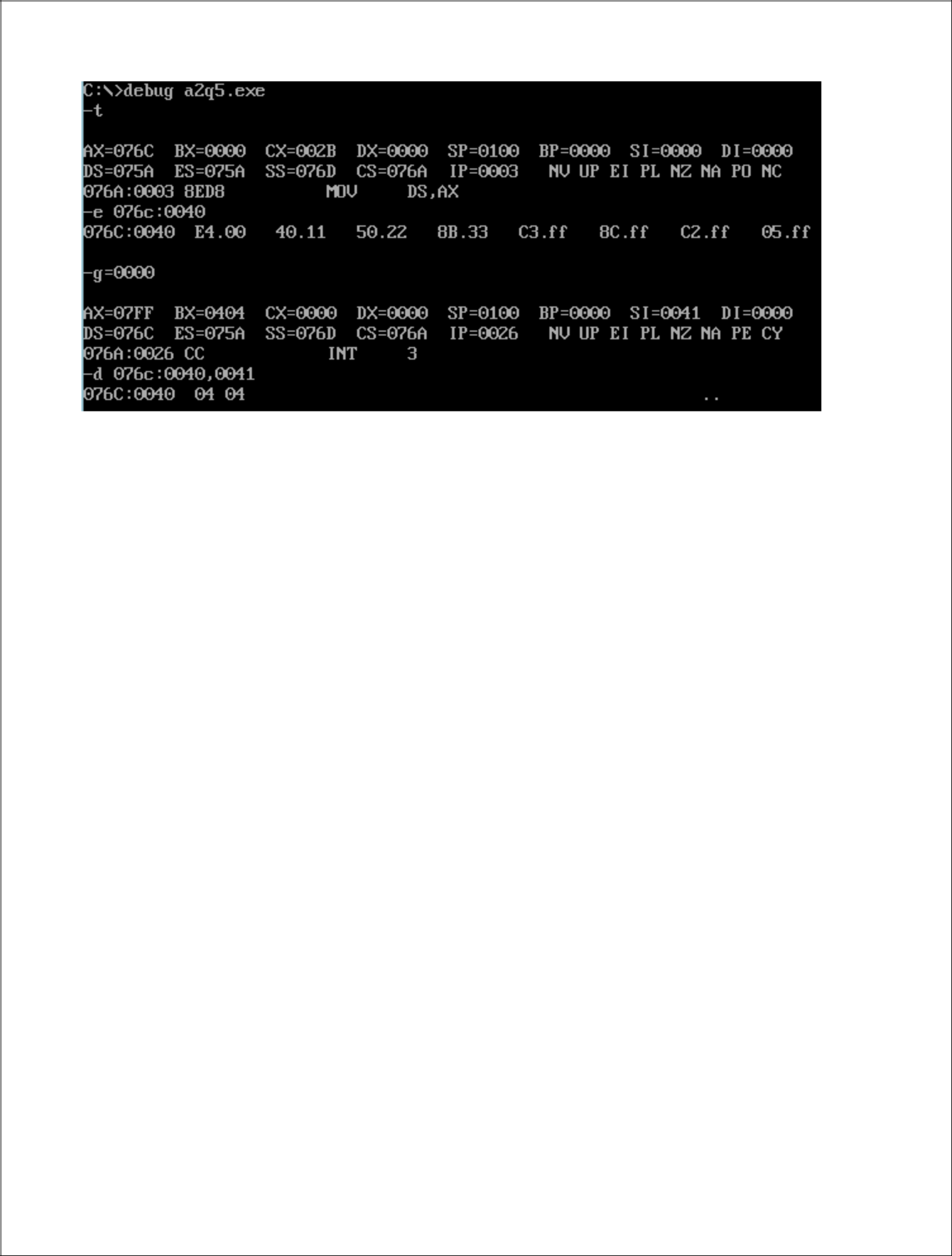
mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to separate the odd and even numbers from a series of 7 data bytes. The starting address of the series is DS: 0030H. Store the even numbers from DS: 0040H and the odd numbers from DS: 0050H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov bx, 0030h

mov si, 0040h ;even

mov di, 0050h ;odd

mov cx, 0007h

l1: mov al, [bx]

ror al, 01h

inc bx

jnc l2

rol al, 01h

mov [di], al ;storing odd in si

inc di

jmp l3

l2: rol al, 01h

mov [si], al

inc si

l3: loop l1

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to convert an 8-bit number stored in DS:0030H into its equivalent ASCII value. Store the converted code from DS: 0050H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov al, [si]

mov ah, al

and al, 0fh

cmp al, 09h

jc l2

add al, 07h

l2: add al, 30h

mov si, 0050h

mov [si], al

inc si

mov al, ah

and al, 0f0h

mov cl, 04h

rol al, cl

cmp al, 09h

jc l3

add al, 07h

l3: add al, 30h

mov [si], al

int 03h

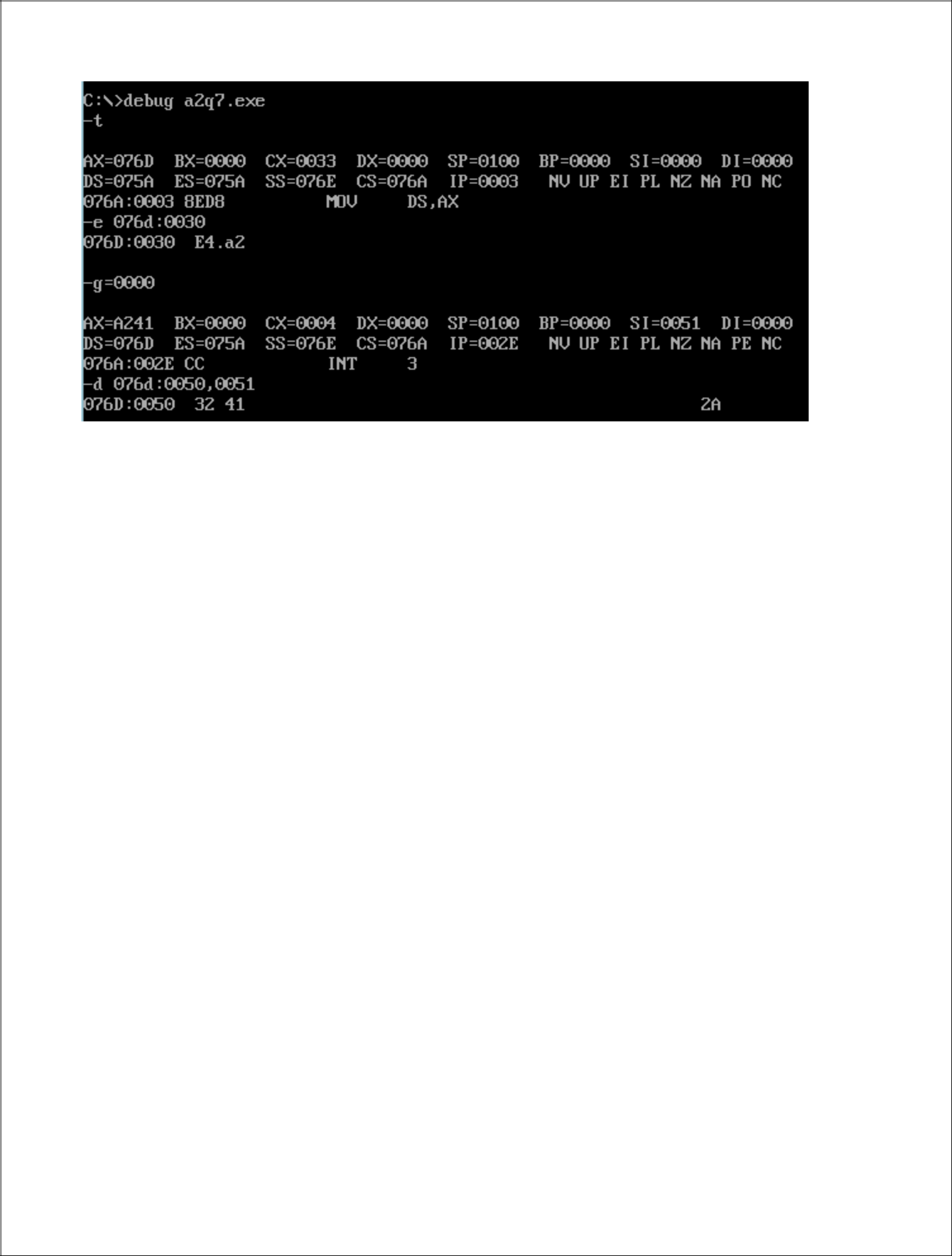
mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Write an Assembly Language Program to find out the square root of a number stored in DS: 0030H. Store the result in DS: 0040H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov al, [si]

mov bl, 01h

mov cl, 00h

l1: sub al, bl

das

add bl, 02h

daa

inc cl

cmp al, 00h

jz l2

jmp l1

l2: mov si, 0040h

mov [si], cl

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

1. Fibonacci series is defined as:

F(i) = F(i-1) + F(i-2); for all i>2 with F(1) = F(2) = 1

Write an Assembly language Program to generate the first ten elements of this sequence and store them from DS: 0030H.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov cx, 000ah

mov al, 01h

mov bl, 01h

mov si, 0030h

l1:

mov [si], al

inc si

mov [si], bl

inc si

add al, bl

daa

xchg al, bl

add al, bl

daa

xchg al, bl

loop l1

int 03h

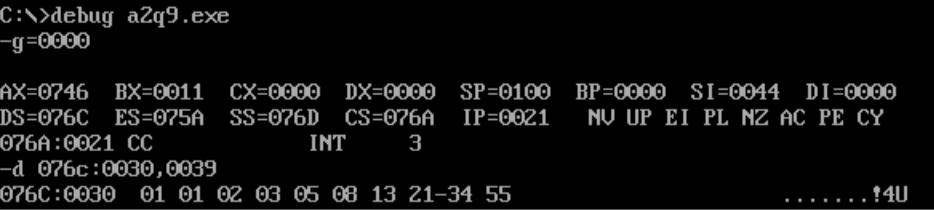
mov ah, 4ch

int 21h

main endp

end main

**Output:**

****

**Assignment 3**

1.Write an Assembly Language Program to find the smallest number from a series of seven data bytes stored from DS: 0030H. Store the smallest number in DS: 0040H.  
  
.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov al, 0ffh

mov cx, 0007h

l1:

cmp al, [si]

jc l2

mov al, [si]

l2:

inc si

loop l1

mov si, 0040h

mov [si], al

int 03h

main endp

end main

2. Write an Assembly Language Program to find the largest number from a series of 7 sixteen-bit numbers stored from DS: 0030H. Store the largest number in DS: 0040H.  
  
.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov al, 00h

mov cx, 0007h

l1:

cmp al, [si]

jnc l2

mov al, [si]

l2:

inc si

loop l1

mov si, 0040h

mov [si], al

int 03h

main endp

end main

3. Write an Assembly Language Program to arrange a series of 7 data bytes stored from DS: 0030H in ascending order.  
  
.model small

.stack 100h

.code

main proc

mov ax, @data

mov ds, ax

mov bl, 06h

l3:

mov si,0030h

mov cl,06h

l1:

mov al,[si]

inc si

cmp al,[si]

jc l2

mov dl,[si]

mov [si],al

dec si

mov [si],dl

inc si

l2:

loop l1

dec bl

cmp bl,00h

jnz l3

int 03h

mov ah,4ch

int 21h

main endp

end main

4. Write an Assembly Language Program to arrange a series of 7 sixteen-bits data stored from DS: 0030H in descending order.  
  
.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov es, ax

mov ds, ax

mov si, 0030h

mov cx, 0006h

l1:

mov si, 0030h

mov bx, cx

l2:

mov ax, [si]

mov dx, [si + 2]

cmp ax, dx

jnc l3

mov [si], dx

mov [si + 2], ax

l3:

add si, 2

dec bx

jnz l2

loop l1

int 03h

mov ah, 4ch

int 21h

main endp

end main

1. Write an Assembly Language program to find the square of a number stored in DS: 0030H using LOOK-UP table. Assume that the LOOK-UP table is stored from DS: 0040H that contains the square of the numbers 0 to 9. Store the square value in DS: 0050H.

|  |  |
| --- | --- |
| DS:0100H | 00 |
| DS:0101H | 01 |
| DS:0102H | 04 |
| DS:0103H | 09 |
| DS:0104H | 16 |
| DS:0105H | 25 |
| DS:0106H | 36 |
| DS:0107H | 49 |
| DS:0108H | 64 |
| DS:0109H | 81 |

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov es, ax

mov ds, ax

mov si, 0030h

mov al, [si]

mov bx,0040h

xlat

mov si,0050h

mov [si],al

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Jadavpur University**

**Session 2024-25, Odd Semester**

**Microprocessor Lab**

**Paper Code: IT/ S/222**

**Assignment 4**

1. Write an Assembly Language Program to add 3 X 3 matrices. Assume the matrices are stored in the form of lists (row wise). First matrix is stored from DS:0030H and the second matrix is stored from DS:0040.Store the result of the addition in the third lists starting from DS:0050H.  
     
   dosseg

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov es,ax

mov ds, ax

mov si,0030h

mov di,0040h

mov bx,0050h

mov cx,0009h

l1:

mov al,[si]

add al,[di]

mov [bx],al

inc di

inc bx

inc si

loop l1

int 03h

mov ah,4ch

int 21h

main endp

end main

2.Write an Assembly Language Program to convert an eight bit binary number stored in DS:0030H into its equivalent BCD number. Stored the result in DS:0040H.

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov ax,0000h

mov dx,0000h

mov si,0030h

mov cl,[si]

l2:

cmp cl,00h

jz l1

dec cl

MOV AL,DL

add al,01h

daa

mov dl,al

mov al,dh

adc al,00h

daa

mov dh,al

jmp l2

l1:

mov si,0040h

mov [si],dx

int 03h

mov ah,4ch

int 21h

main endp

end main

1. Write an Assembly program to convert a BCD number stored in DS:0030H into its equivalent hexadecimal number. Stored the result in DS:0040H.

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov si,0030h

mov al,[si]

mov bl,al

AND AL,0F0H

mov cl,04h

ror al,cl

mov dl,0Ah

mul dl

mov dx,ax

mov al,bl

and al,0fh

mov ah,00h

add ax,dx

mov si,0040h

mov [si],ax

int 03h

mov ah,4ch

int 21h

main endp

end main

1. Write an Assembly program to convert a binary number stored in DS:0030H into its equivalent gray code. Stored the result in DS:0040H.

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov si,0030h

mov ax,0000h

mov al,[si]

mov bl,al

clc

rcr al,01h

xor al,bl

mov si,0040h

mov [si],ax

int 03h

mov ah,4ch

int 21h

main endp

end main

1. Write an Assembly program to find the factorial of a number stored in DS:0030H. Stored the result in DS:0040H.

.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax

mov ax,0000h

mov bx,0000h

mov cx,0001h

mov si,0030h

mov bl,[si]

mov al,bl

l2:

cmp bl,00h

jz l1

dec bl

cmp bl,00h

jz l1

MUL BX

mov cx,ax

jmp l2

l1:

mov si,0040h

mov [si],cx

int 03h

mov ah,4ch

int 21h

main endp

end main