ASM-LAB Assignment 4



NAME: Swarnendu Banerjee

ROLL: 002311001016

DEPARTMENT: IT-A1

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.

Code

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

;mov es, ax

mov si, 0030h

mov bx, 0040h

mov di, 0050h

mov cx, 0009h

11:

mov al, [si]

add al, [bx]

mov [di], al

inc di

inc si

inc bx

loop I1

int 03h

mov ah, 4ch

int 21h

main endp

end main

Output

```
C:\>debug a4q1.exe
4X=076C
         BX=0000
                  CX=0021
                           DX=0000
                                     SP=0100
                                              BP=0000 SI=0000 DI=0000
DS=075A ES=075A
                  SS=076D
                                     IP=0003
                                               NV UP EI PL NZ NA PO NC
                           CS=076A
076A:0003 8ED8
                        MOV
                                DS,AX
-e 076c:0030
076C:0030 3D.1
                   FF.2
                           FF.3
                                    74.4
                                            03.5
                                                    E9.6
                                                            ED.7
                                                                     00.8
076C:0038 C4.9
e 076c:0040
076C:0040 E4.1
                   40.2
                           50.3
                                    8B.4
                                            C3.5
                                                    80.6
                                                            C2.7
                                                                     05.8
0760:0048 00.9
g=0000
AX=0712
         BX=0049
                  CX=0000
                           DX=0000 SP=0100
                                              BP=0000 SI=0039 DI=0059
DS=076C
                  SS=076D
                                               NV UP EI PL NZ NA PO NC
         ES=075A
                           CS=076A
                                     IP=001C
076A:001C CC
                        INT
                                3
-d 076c:0050,0059
076C:0050 02 04 06 08 0A 0C 0E 10-12 17
                                                               . . . . . . . . . .
```

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.

Code

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov bl, [si]

mov ax, 0000h

mov dx, 0000h

cmp bl, 00h

jz 12

11:

add ax, 01h

daa

adc dl, 00h

```
dec bl
cmp bl, 00h
jnz l1
l2: mov si, 0040h
mov [si], dl
inc si
```

int 03h mov ah, 4ch int 21h

mov [si], ax

main endp end main

Output:

```
AX=076C
        BX=0000
                  CX=002B
                           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.ff
 g=0000
X=0055
         BX=0000
                  CX=002B
                           DX=0002
                                    SP=0100
                                             BP=0000 SI=0041 DI=0000
DS=076C
        ES=075A
                  SS=076D CS=076A
                                    IP=0026
                                              NU UP EI PL NZ NA PE NC
076A:0026 CC
                        INT
                                3
-d 076c:0040,0041
076C:0040 02 55
                                                              .U
-e 076c:0030
076C:0030 FF.a4
 0000 g
         BX=0000
                  CX=002B
                           DX=0001
                                    SP=0100
                                             BP=0000 SI=0041 DI=0000
DS=076C
       ES=075A
                  SS=076D
                          CS=076A
                                    IP=0026
                                              NU UP EI PL NZ NA PE NC
076A:0026 CC
                        INT
                                3
-d 076c:0040,0041
076C:0040 01 64
                                                              . d
```

3. Write an Assembly program to convert a BCD number stored in DS:0030H into its equivalent hexadecimal number. Stored 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, 00h

cmp al, 00h jz l2

I1: sub al, 01h das inc bl cmp al, 00h jz l2 jmp I1

I2: mov si, 0040h mov [si], bl

int 03h mov ah, 4ch int 21h

main endp end main

Output:

```
C:\>debug a4q3.exe
        BX=0000
                 CX=0021
                           DX=0000
AX=076C
                                    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
-е 076c:0030
076C:0030 96.55
g=0000
AX=0700
        BX=0037
                 CX=0021
                           DX=0000
                                    SP=0100
                                             BP=0000 SI=0040 DI=0000
DS=076C
        ES=075A
                          CS=076A
                                              NU UP EI PL ZR NA PE NC
                 SS=076D
                                    IP=001C
076A:001C CC
                        INT
                                3
-d 076c:0040,0040
0760:0040 37
                                                             7
```

4. Write an Assembly program to convert a binary number stored in DS:0030H into its equivalent gray code. Stored 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, al

ror bl, 01h xor al, bl

mov si, 0040h mov [si], al

int 03h mov ah, 4ch int 21h

main endp end main

Output:

```
C:\>debug a4q4.exe
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
                                              NV UP EI PL NZ NA PO NC
076A:0003 BED8
                        MOV
                                DS,AX
-е 076ь:0030
076B:0030 0E.04
g=0000
AX=0706
         BX=0002
                  CX=001A
                           DX=0000
                                    SP=0100 BP=0000 SI=0040 DI=0000
DS=076B
        ES=075A
                  SS=076C
                           CS=076A
                                    IP=0015
                                              NU UP EI PL NZ NA PE NC
076A:0015 CC
                        INT
                                3
-d 076b:0040,0040
076B:0040 06
```

5. Write an Assembly program to find the factorial 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 bl, [si]
mov bh, 00h
mov ax, 0001h

cmp bl, 00h
jz l2
```

11:

jz l2

cmp bl, 01h

mul bx dec bl cmp bl, 01h jz l2 jmp l1

I2: mov si, 0040h mov [si], dh inc si mov [si], dl inc si mov [si], ah inc si mov [si], al

int 03h mov ah, 4ch int 21h

main endp end main

Output:

```
C:\>debug a4q5.exe
AX=076C
        BX=0000
                 CX=002D
                          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 BED8
                       MOV
                               DS,AX
-e 076c:0030
0760:0030 07.07
q=0000
AX=13B0
        BX=0001
                 CX=002D
                          DX=0000
                                   SP=0100
                                             BP=0000 SI=0043
                                                               DI=0000
DS=076C
        ES=075A
                 SS=076D
                          CS=076A
                                    IP=0028
                                             NV UP EI PL NZ NA PO NC
076A:0028 CC
                        INT
                               3
-d 076c:0040,0043
076C:0040 00 00 13 B0
```