Name: ANUSKA NATH

Roll Number: 002311001003

Section: A1

Dept: INFORMATION TECHNOLOGY

Year: UG2 Sem 1

ASM lab assignment - 2

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.

```
.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
```

```
::\>debug a2q1.exe
                 CX=0022
                          DX=0000
                                   SP=0100
                                            BP=0000 SI=0000 DI=0000
                         CS=076A
DS=075A ES=075A
                                   IP=0003
                                              NU UP EI PL NZ NA PO NC
076A:0003 BED8
                                DS, AX
-е 076c:0030
976C:0030 11.00
                  22.11
                          33.55
                                  55.22
                                           44.55
                                                  55.33
                                                           22.55
                                                                   33.22
                          DX=0000
                                   SP=0100
                                            BP=0000 SI=0040 DI=0038
                 SS=076D CS=076A
                        INT
d 076c:0040,0040
```

int 03h

mov si, 0040h

mov [si], bl

11:

scasb

inz 12

inc bl 12: loop 11 mov ah, 4ch int 21h main endp

end main

2. 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.

```
.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

11:
scasb
```

mov [si], di
add si, 0002h
inc di
12:
loop 11

mov ah, 4ch int 21h

jnz 12 dec di

main endp end main

```
:\>debug a2q2.exe
                                    SP=0100
DS=075A
        ES=075A
                                     IP=0003
976A:0003 BED8
e 076c:0030
                           55.55
                                    22.33
976C:0030 55.00
                                            55.22
                                                    22.55
                                                                     33.55
        BX=0000
                           DX=0000
                                    SP=0100
                                              BP=0000 SI=0046 DI=0038
DS=076C
        ES=076C
                  SS=076D
                          CS=076A
                                     IP=001F
                                               NU UP EI PL NZ NA PO NC
076A:001F CC
                        INT
d 076c:0040,0045
976C:0040 32 00 35 00 37 00
```

3. 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.

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

```
main proc
```

int 21h

```
\>debug a2q3.exe
mov ax, @data
mov ds, ax
                           BX=0000
                                     CX=002A
                                               DX=0000
                                                         SP=0100
                                                                  BP=0000 SI=0000
                                                                                      DI=0000
mov es, ax
                    =075A
                           ES=075A
                                     SS=076D
                                               CS=076A
                                                         IP=0003
                                                                   NU UP EI PL NZ NA PO NC
                                            MOU
                                                    DS, AX
                  )76A:0003 8ED8
                    076c:0030
mov si, 0030h
                  076C:0030 3D.05
                                      FF.11
                                               FF.22
                                                        74.33
                                                                03.44
mov di, 0040h
                  -е 076c:0040
                  076C:0040 E4.05
                                      40.11
                                               50.22
                                                        8B.33
                                                                C3.44
mov cl, [si]
                   00000 = p
mov ch, 00h
cld
                  X=FFFF
                                               DX=0000
                           BX=0050
                                     CX=0000
                                                        SP=0100
                                                                  BP=0000 SI=0035 DI=0045
                                     SS=076D
                           ES=076C
                                               CS=076A
                                                         IP=0025
                                                                   NU UP EI PL ZR NA PE NC
                   S=076C
                   )76A:0025 CC
                                            INT
                                                    3
11:
                    076c:0050,0051
cmpsb
jnz 12
```

```
loop 11
                  ::\>debug a2q3.exe
mov ax, 0ffffh
                  4X=076C
                           BX=0000
                                     CX=002A
                                              DX=0000
                                                        SP=0100
                                                                  BP=0000 SI=0000 DI=0000
jmp 13
                                                        IP=0003
                  DS=075A
                           ES=075A
                                     SS=076D
                                              CS=076A
                                                                   NU UP EI PL NZ NA PO NC
                                           MOV
                                                    DS, AX
                  976A:0003 8ED8
                  e 076c:0030
12:
                                              22.22
                  976C:0030 05.05
                                      06.06
                                                       33.33
                                                                44.44
mov ax, 01111h
                  e 076c:0040
                  076C:0040 05.05
                                      11.11
                                              22.22
                                                       33.33
                                                                44.44
13:
                  g=0000
mov bx, 0050h
mov [bx], ax
                  AX=1111
                           BX=0050
                                     CX=0004
                                              DX=0000
                                                        SP=0100
                                                                  BP=0000 SI=0032
                                                                                     DI=0042
                  DS=076C
                           ES=076C
                                     SS=076D
                                              CS=076A
                                                        IP=0025
                                                                   NU UP EI NG NZ NA PE CY
                  076A:0025 CC
                                            INT
                                                    3
int 03h
                  -d 076c:0050,0051
                  0760:0050
                             11 11
mov ah, 4ch
```

main endp end main

4. 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.

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

```
>debug a2q4.exe
mov ax, 0005h
                      1X=076D
                              BX=0000
                                        CX=0038
                                                 DX=0000
                                                           SP=0100
                                                                    BP=0000 SI=0000 DI=0000
mov si, 0030h
                      )S=075A
                              ES=075A
                                        SS=076E
                                                 CS=076A
                                                           IP=0003
                                                                     NV UP EI PL NZ NA PO NC
                      976A:0003 8ED8
                                              MOV
                                                       DS,AX
mov di, 0030h
                      e 076d:0030
                                                 50.3
                                                          8B.2
add di, ax
                      076D:0030 E4.1
                                         40.2
                                                                  C3.1
dec di
                      g=0000
                                                 DX=0000
                     AX=FFFF
                              BX=0040
                                        CX=0000
                                                           SP=0100
                                                                    BP=0000 SI=0030 DI=0034
mov bl, 02h
                                        SS=076E
                                                 CS=076A
                                                                     NU UP EI PL ZR NA PE NC
                     DS=076D
                              ES=076D
                                                           IP=0033
div bl
                     076A:0033 CC
                                              INT
                                                       3
                      d 076d:0040,0041
mov cl, al
                      76D:0040 FF FF
mov ch, 00h
```

```
11:
mov al, [si]
mov bl, [di]
cmp al, bl
jnz 12
loop 11
mov ax, 0ffffh
jmp 13
```

mov ax, 01111h

12:

```
::\>debug a2q4.exe
AX=076D
        BX=0000
                  CX=0038
                           DX=0000
                                    SP=0100
                                              BP=0000 SI=0000 DI=0000
                                    IP=0003
                  SS=076E
                                               NV UP EI PL NZ NA PO NC
DS=075A ES=075A
                           CS=076A
076A:0003 8ED8
                                DS,AX
                        MOV
-e 076d:0030
076D:0030 01.1
                   02.2
                           03.3
                                   02.4
                                            01.5
g=0000
AX=1111
        BX=0040
                  CX=000Z
                           DX=0000
                                    SP=0100
                                              BP=0000 SI=0030 DI=0034
DS=076D
        ES=076D
                  SS=076E
                           CS=076A
                                     IP=0033
                                               NU UP EI NG NZ AC PE CY
976A:0033 CC
                        INT
                                3
-d 076d:0040,0041
976D:0040 11 11
```

13:
mov bx, 0040h
mov [bx], ax
int 03h
mov ah, 4ch
int 21h
main endp
end main

5. 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.

```
.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
```

```
11: mov al, [si]
rol al, 01h
                                                                BP=0000 SI=0000 DI=0000
inc si
                         BX=0000
                                   CX=002B
                                             DX=0000
                                                       SP=0100
                                                       IP=0003
                         ES=075A
                                   SS=076D
                                             CS=076A
                                                                  NU UP EI PL NZ NA PO NC
jc 12
                )76A:0003 8ED8
                                                  DS, AX
inc bh
                e 076c:0040
                976C:0040 E4.00
                                    40.11
                                             50.22
                                                     8B.33
                                                              C3.ff
                                                                       8C.ff
                                                                                CZ.ff
                                                                                         05.ff
jmp 13
12: inc bl
                g=0000
13: loop 11
                         BX=0404
                                   CX=0000
                                            DX=0000
                                                       SP=0100
                                                                BP=0000 SI=0041
               DS=076C
                         ES=075A
                                   SS=076D
                                            CS=076A
                                                       IP=0026
                                                                  NU UP EI PL NZ NA PE CY
mov si, 0040h
                                                  3
                                          INT
               076A:0026 CC
                d 076c:0040,0041
mov [si], bh
inc si
```

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

6. 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.

```
.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
11: mov al, [bx]
ror al, 01h
inc bx
jnc 12
rol al, 01h
mov [di], al ;storing odd in si
inc di
jmp 13
12: rol al, 01h
mov [si], al
inc si
13: loop 11
```

```
::>debug a2q6.exe
int 03h
                                                                             DI=0000
            X=076C
                     BX=0000
                              CX=00ZE
                                       DX=0000
                                                 SP=0100
                                                          BP=0000 SI=0000
mov ah, 4ch
                    ES=075A
           DS=075A
                              SS=076D
                                       CS=076A
                                                 IP=0003
                                                           NU UP EI PL NZ NA PO NC
int 21h
            076A:0003 8ED8
                                    MOV
                                             DS, AX
            e 076c:0030
            076C:0030 3D.01
                               FF.02
                                       FF.03
                                                74.04
                                                        03.05
                                                                 E9.06
                                                                         ED.07
main endp
end main
            g=0000
                     BX=0037
                              CX=0000
                                       DX=0000
                                                 SP=0100
                                                          BP=0000 SI=0043
           AX=0707
                                                                             DI=0054
            )S=076C
                                                           NU UP EI PL NZ NA PO CY
                    ES=076C
                              SS=076D
                                       CS=076A
                                                 IP=0029
            076A:0029 CC
            d 076c:0040
                      02 04 06 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83
            76C:0040
                       01 03 05 07 86 FA FE 50-E8 17 73 83 C4 06 8B B6
```

7. 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.

```
.model small
.stack 100h
.data
.code
main proc
mov ax, @data
mov ds, ax
mov si, 0030h
mov ah, al
and al, 0fh
cmp al, 09h
jc 12
add al, 07h
12: 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

```
:\>debug a2q7.exe
jc 13
add al, 07h
13: add al, 30h AX=076D
                                                              BP=0000 SI=0000
                      BX=0000
                                CX=0033
                                          DX=0000
                                                    SP=0100
                                                                                  DI=0000
             DS=075A
                      ES=075A
                                SS=076E
                                          CS=076A
                                                    IP=0003
                                                               NU UP EI PL NZ NA PO NC
             076A:0003 8ED8
                                       MOV
                                                DS, AX
mov [si], al
              e 076d:0030
             076D:0030 E4.a2
int 03h
              g=0000
mov ah, 4ch
int 21h
                       BX=0000
                                CX=0004
                                          DX=0000
                                                    SP=0100
                                                              BP=0000
             DS=076D
                      ES=075A
                                SS=076E
                                          CS=076A
                                                     IP=002E
                                                               NU UP EI PL NZ NA PE NC
             976A:00ZE CC
                                       INT
                                                3
main endp
               076d:0050,0051
end main
                                                                                2A
             976D:0050 32 41
```

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

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

main proc

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

```
<>debug a2q8.exe
mov al, [si]
mov bl, 01h
                                                                 BP=0000 SI=0000 DI=0000
                          BX=0000
                                   CX=0027
                                                       SP=0100
                1X=076C
                                             DX=0000
mov cl, 00h
                                                                  NV UP EI PL NZ NA PO NC
                DS=075A
                         ES=075A
                                   SS=076D
                                             CS=076A
                                                       IP=0003
                976A:0003 BED8
                                                   DS, AX
                 e 076c:0030
11: sub al, bl
                976C:0030 3D.49
das
                 g=0000
add bl, 02h
daa
                          BX=000F
                                   CX=0007
                                             DX=0000
                                                       SP=0100
                                                                 BP=0000 SI=0040 DI=0000
                DS=076C
                          ES=075A
                                             CS=076A
                                                       IP=0022
                                                                  NU UP EI PL ZR NA PE NC
inc cl
                                   SS=076D
                976A:002Z CC
                                          INT
                                                   3
cmp al, 00h
                 d 076c:0040,0040
jz 12
                976C:0040 07
jmp 11
```

12: mov si, 0040h mov [si], cl

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

9. 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.

.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

11:

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 11

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

```
C:\>debug aZq9.exe

-g=0000

AX=0746 BX=0011 CX=0000 DX=0000 SP=0100 BP=0000 SI=0044 DI=0000

DS=076C ES=075A SS=076D CS=076A IP=0021 NV UP EI PL NZ AC PE CY

076A:0021 CC INT 3

-d 076c:0030,0039

076C:0030 01 01 02 03 05 08 13 21-34 55 .....!4U
```