

**Name: Dripta Patra**  
**Roll No: 002311001006**  
**Section: A1**  
**Dept: IT**  
**Year: UG2**  
**Semester: 1st**

## **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.

**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:**

```

C:\>debug a2q1.exe
-t

AX=076C BX=0000 CX=0022 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 11.00  22.11  33.55  55.22  44.55  55.33  22.55  33.22

-g=0000

AX=0755 BX=0003 CX=0000 DX=0000 SP=0100 BP=0000 SI=0040 DI=0038
DS=076C ES=076C SS=076D CS=076A IP=001D  NV UP EI PL NZ NA PE NC
076A:001D CC          INT     3
-d 076c:0040,0040
076C:0040 03

```

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.

**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:**

```

C:\>debug a2q2.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 55.00  01.11  55.55  22.33  55.22  22.55  11.44  33.55

-g=0000

AX=0755 BX=0000 CX=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     3
-d 076c:0040,0045
076C:0040 32 00 35 00 37 00                2.5.7.

```

3. Write an Assembly Language Program to compare two strings. The first string is stored from memory location DS: 0030H and the second string 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:

```
C:\>debug a2q3.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 8EDB      MOU      DS,AX
-e 076c:0030
076C:0030 3D.05      FF.11      FF.22      74.33      03.44
-e 076c:0040
076C:0040 E4.05      40.11      50.22      8B.33      C3.44
-g=0000
AX=FFFF BX=0050 CX=0000 DX=0000 SP=0100 BP=0000 SI=0035 DI=0045
DS=076C ES=076C SS=076D CS=076A IP=0025  NU UP EI PL ZR NA PE NC
076A:0025 CC      INT      3
-d 076c:0050,0051
076C:0050 FF FF      ..
```

```
C:\>debug a2q3.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 8EDB      MOU      DS,AX
-e 076c:0030
076C:0030 05.05      06.06      22.22      33.33      44.44
-e 076c:0040
076C:0040 05.05      11.11      22.22      33.33      44.44
-g=0000
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
-d 076c:0050,0051
076C:0050 11 11      ..
```

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 si,0030h
mov di, 0034h
mov cx,02h
l2 :
mov ah,[si]
mov bh, [di]
cmp ah,bh
jnz l1
inc SI
dec DI
```

```
mov ax,0FFFFh
mov bx,0040h
mov [bx],ax
loop l2
```

```

jmp 13

l1:
mov ax,1111h
mov bx,0040h
mov [bx],ax
l3:
int 03h
mov ah,4ch
int 21h
main endp
end main
main endp
end main

```

Output:

```

C:\>debug a2q4.exe
-t
AX=076D BX=0000 CX=0038 DX=0000 SP=0100 BP=0000 SI=0000 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
-e 076d:0030
076D:0030 E4.1      40.2      50.3      8B.2      C3.1

-g=0000

AX=FFFF BX=0040 CX=0000 DX=0000 SP=0100 BP=0000 SI=0030 DI=0034
DS=076D ES=076D SS=076E CS=076A IP=0033  NU UP EI PL ZR NA PE NC
076A:0033 CC          INT     3
-d 076d:0040,0041
076D:0040 FF FF
..

```

```

C:\>debug a2q4.exe
-t
AX=076D BX=0000 CX=0038 DX=0000 SP=0100 BP=0000 SI=0000 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
-e 076d:0030
076D:0030 01.1      02.2      03.3      02.4      01.5

-g=0000

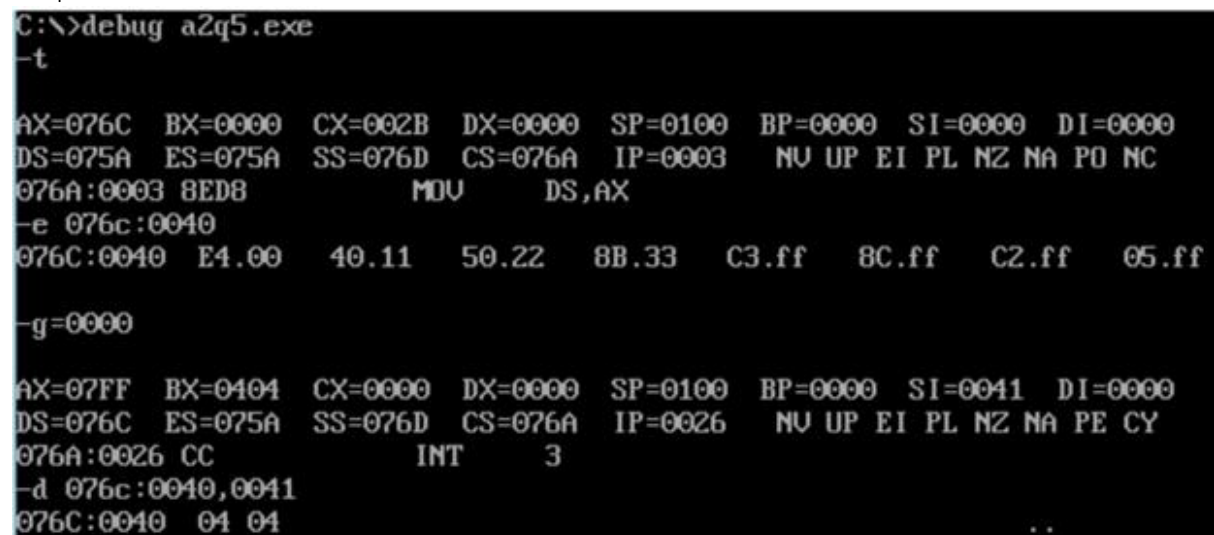
AX=1111 BX=0040 CX=0002 DX=0000 SP=0100 BP=0000 SI=0030 DI=0034
DS=076D ES=076D SS=076E CS=076A IP=0033  NU UP EI NG NZ AC PE CY
076A:0033 CC          INT     3
-d 076d:0040,0041
076D:0040 11 11
..

```

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.

**Code:**

```
.model small
.stack 100h
.data
.code
main proc
mov ax, @data
mov ds, ax
mov bx, 0000h
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:**

```
C:\>debug a2q5.exe
-t
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  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 E4.00  40.11  50.22  8B.33  C3.ff  8C.ff  C2.ff  05.ff
-g=0000
AX=07FF BX=0404 CX=0000 DX=0000 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 CY
076A:0026 CC          INT     3
-d 076c:0040,0041
076C:0040 04 04
```

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
mov di,0050h
mov cx,0007h
l3:
mov al,[bx]
ror al,01h
jnc l1
rol al,01h
mov [di],al
inc di
jmp l2
l1:
rol al,01h
mov [si],al
inc si
l2:
inc bx
loop l3
```

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

Output:

```
C:\>debug aZq6.exe
-t

AX=076C  BX=0000  CX=002E  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.01  FF.02  FF.03  74.04  03.05  E9.06  ED.07

-g=0000

AX=0707  BX=0037  CX=0000  DX=0000  SP=0100  BP=0000  SI=0043  DI=0054
DS=076C  ES=076C  SS=076D  CS=076A  IP=0029  NU UP EI PL NZ NA PO CY
076A:0029 CC          INT     3
-d 076c:0040
076C:0040 02 04 06 8B C3 8C C2 05-0C 00 52 50 EB C1 48 83  ....R
076C:0050 01 03 05 07 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ....P..s
```

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.

**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:**



```

C:\>debug a2q7.exe
-t

AX=076D BX=0000 CX=0033 DX=0000 SP=0100 BP=0000 SI=0000 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
-e 076d:0030
076D:0030 E4.a2

-g=0000

AX=A241 BX=0000 CX=0004 DX=0000 SP=0100 BP=0000 SI=0051 DI=0000
DS=076D ES=075A SS=076E CS=076A IP=002E  NU UP EI PL NZ NA PE NC
076A:002E CC          INT     3
-d 076d:0050,0051
076D:0050 32 41                      2A

```

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.

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 l1l2: mov si, 0040h
mov [si], cl
int 03h
mov ah, 4ch
int 21h
main endp
end main

```

Output:

```

C:\>debug a2q8.exe
-t

AX=076C BX=0000 CX=0027 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.49

-g=0000

AX=0700 BX=000F CX=0007 DX=0000 SP=0100 BP=0000 SI=0040 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0022  NU UP EI PL ZR NA PE NC
076A:0022 CC          INT     3
-d 076c:0040,0040
076C:0040 07

```

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.

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:

```

C:\>debug a2q9.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  NU 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

```

