

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

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
```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
0 Severe Errors

C:\>link a3q1.obj:

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

C:\>debug a1q3.exe
-t

AX=076B BX=0000 CX=0018 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076b:0030
076B:0030 00.1    52.2    50.3    E8.4    EA.5    48.6    83.7    C4.

-g=0000

AX=076B BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0038 DI=0048
DS=076B ES=076B SS=076C CS=076A IP=0013  NU UP EI PL NZ NA PO NC
076A:0013 CC          INT     3
-d 076b:0040,0040
076B:0040 01

```

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
```

```
C:\>debug a3q2.exe
-t
AX=076B BX=0000 CX=001C DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-t
AX=076B BX=0000 CX=001C DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=0005  NU UP EI PL NZ NA PO NC
076A:0005 BE3000      MOV     SI,0030
-e 076b:0030
076B:0030 00.11 52.11 50.22 EB.22 EA.33 48.33 83.44 C4.44
076B:0038 04.55 50.55 EB.66 7B.66 0E.77 83.77 C4.
-g=0000
AX=0744 BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0040 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=001B  NU UP EI PL NZ NA PO CY
076A:001B CC          INT     3
-d 076b:0040,0041
076B:0040 44 FF          D.
```

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
endmain

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG
C:\>link a3q2.obj;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
C:\>debug a3q2.exe
-t
AX=076C BX=0000 CX=0029 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.12  FF.43  FF.9   74.87  03.55  E9.32  ED.21  00.
-g=0000
AX=0755 BX=0000 CX=0000 DX=0021 SP=0100 BP=0000 SI=0036 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0024  NV UP EI PL ZR NA PE NC
076A:0024 CC          INT     3
-d 076c:0040,0046
076C:0040 E4 40 50 8B C3 8C C2          .@P....
-d 076c:0030,0036
076C:0030 09 12 21 32 43 55 87      ..!2CU.

```

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
endmain

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG
0 Severe Errors

C:\>link a3q2.obj;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

C:\>debug a3q2.exe
-t

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  NU UP EI PL NZ NA PO NC
076A:0003 8EC0          MOV     ES,AX
-e 076c:0030
076C:0030 3D.12 FF.54 FF.66 74.79 03.1 E9.23 ED.44 00.

-g=0000

AX=7966 BX=0000 CX=0000 DX=5EC4 SP=0100 BP=0000 SI=0032 DI=0000
DS=076C ES=076C SS=076D CS=076A IP=0028  NU UP EI PL ZR NA PE NC
076A:0028 CC          INT     3
-d 076c:0030,0036
076C:0030 66 79 C4 5E 12 54 8A          fy.^.T.

```

- 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

```

```

.model small
.stack 100h
.data
.code

main proc
    mov ax, @data
    mov ds, ax

    mov si, 0030h

    mov al, [si]
    mov bx, 0100h
    xlat

    mov si, 0040h
    mov [si], al

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

```

```
C:\>debug assn3q5.exe
```

```
-t
```

```
AX=076B BX=0000 CX=001B DX=0000 SP=0100 BP=0000 SI=0000 DI=0000  
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC  
076A:0003 8ED8          MOV     DS,AX
```

```
-e 076b:0100
```

```
076B:0100 FF.00  50.01  E8.04  89.09  69.16  83.25  6B.36  07.49
```

```
076B:0108 00.64  00.81
```

```
-e 076b:0030
```

```
076B:0030 00.05
```

```
-g=0000
```

```
AX=0725 BX=0100 CX=001B DX=0000 SP=0100 BP=0000 SI=0040 DI=0000  
DS=076B ES=075A SS=076C CS=076A IP=0013  NU UP EI PL NZ NA PO NC  
076A:0013 CC          INT     3
```

```
-d 076b:0040,0040
```

```
076B:0040 25          %
```