# Assignment:1

**1. Write a program to perform addition of two 8-bit numbers.**

```
MOV AH,09H
MOV AL,10H
ADD AH,AL
HLT
```

**2. Write a program to perform subtraction of two 8-bit numbers**.

```
MOV AH,09H
MOV AL,10H
cmp ah,al
jc label
sub ah,al
HLT
label:
sub ah,al
neg ah
```

**3. Write a program to perform multiplication of two 8-bit numbers**

```
MOV AH,12H
MOV AL,04H
MUL Al
HLT
```

**4. Write a program to perform division of two 8-bit numbers.**

```
MOV AX, 12H
MOV BL, 04H
DIV BL
HLT
```

**5. Write a program to interchange values of two variables.**

```
 MOV AX, 5
MOV BX, 10
MOV CX, AX
MOV AX, BX
MOV BX, CX
HLT
```

**6. Write a program to print an array.**

```
include 'emu8086.inc'
.model small
.data
n1 DB 5 dup(?)
.CODE
main proc
mov ax,@data
mov ds,ax
mov si,offset n1
mov cx,05H
print 'Input:->'
lab1:
mov ah,01h
int 21h
mov [si],al
mov dl,32
mov ah,02H
int 21H
inc si
loop lab1
mov dl,10
mov ah,02
int 21h
mov dl,13
mov ah,02
int 21h
mov cx,0005H
mov si,offset n1
print 'Output:->'
lab2:
mov dl,[si]
mov ah,02H
int 21h
mov dl,32
mov ah,02H
int 21h
inc si
loop lab2
mov ah,4ch
int 21h
main endp
end
```

### 7. Write a program to print the reverse of an array.

```
include 'emu8086.inc'
.model small
.data
n1 DB 5 dup(?)
.CODE
main proc
mov ax,@data
mov ds,ax
mov si,offset n1
mov cx,05H
print 'Input:->'
lab1:
mov ah,01h
int 21h
mov [si],al
mov dl,32
mov ah,02H
int 21H
inc si
loop lab1
mov dl,10
mov ah,02
int 21h
mov dl,13
mov ah,02
int 21h
mov cx,0005H
mov si,offset n1
add si,cx
dec si
print 'Output:->'
lab2:
mov dl,[si]
mov ah,02H
int 21h
mov dl,32
mov ah,02H
int 21h
dec si
loop lab2
mov ah,4ch
int 21h
main endp
 end
```

### 8. Write a program to find maximum number from given array of 16-bit numbers

```
include 'emu8086.inc'
.model small
.data
n1 DB 5 DUP(?)
max_val DB ?
min_val DB ?
.CODE
main proc
 mov ax, @data
 mov ds, ax
 mov si, offset n1
 mov cx, 05H
 print 'Input:->'
lab1:
 mov ah, 01h
 int 21h
 mov [si], al
 mov dl, 32
 mov ah, 02H
 int 21H
 inc si
 loop lab1
 mov si, offset n1
 mov al, [si]
 mov max_val, al
 mov min_val, al
 mov cx, 05H
 mov si, offset n1
lab2:
 mov al, [si]
 cmp al, max_val
 jbe check_min
 mov max_val, al
check_min:
 cmp al, min_val
 jae next_element
 mov min_val, al
next_element:
 inc si
 loop lab2
 mov cx, 05H
 mov si, offset n1
 print 'Output:->'
lab3:
 mov dl, [si]
 mov ah, 02H
 int 21h
 mov dl, 32
 mov ah, 02H
 int 21h
 inc si
 loop lab3
 mov dl, 10
 mov ah, 02H
 int 21H
 mov dl, 13
 mov ah, 02H
 int 21H
 print 'Max: '
 mov dl, max_val
 mov ah, 02H
 int 21H
 mov dl, 10
 mov ah, 02H
 int 21H
 mov dl, 13
 mov ah, 02H
 int 21H
 print 'Min: '
 mov dl, min_val
 mov ah, 02H
 int 21H
 mov ah, 4ch
 int 21h
main endp
end main
```