Experiment No. 8

Name – Mehatab Mahibub Sanadi
Roll No. – CO2056

```
section .data
        welmsg db 10,'Multiplication using add and shift method',10
        welmsg_len equ $-welmsg
        nummsg db 10,'Enter two digits of Number::'
        nummsg_len equ $-nummsg
        resmsg db 10,'Multiplication of elements::'
        resmsg_len equ $-resmsg
        nline db 10,'',10
        nline_len equ $-nline

section .bss
        numascii resb 03
        multi1 resb 02
        multi2 resb 02
        resl resb 02
        dispbuff resb 04

%macro dispmsg 2
        mov eax,04
        mov ebx,01
        mov ecx,%1
        mov edx,%2
        int 80h
        %endmacro

%macro accept 2
        mov eax,03
        mov ebx,0
        mov ecx,%1
        mov edx,%2
        int 80h
        %endmacro

section .text
global _start
_start:
        dispmsg welmsg,welmsg_len
        dispmsg nummsg,nummsg_len

        accept numascii,3
        call atoh
        mov [multi1],bl

        dispmsg nummsg,nummsg_len
```

```
                accept numascii,3
                call atoh
                mov [multi2],bl

                mov ah,00h
                mov al,[multi1]
                mov bl,[multi2]
                mov cl,08h

addup:
                rcr bl,01
                jnc next1

                add [resl],ax

next1:   shl ax,01
            loop addup


                dispmsg resmsg,resmsg_len

                mov bx,[resl]
                call htoa

                dispmsg nline,nline_len
                mov eax,01 ;Exit
                mov ebx,00
                int 80h

atoh:                   ;i/p proc
                mov bl,0
                mov ecx,02
                mov esi,numascii
                up1:
                rol bl,04
                mov al,[esi]
                cmp al,39h
                jbe skip1
                sub al,07h
                skip1: sub al,30h
                add bl,al
                inc esi
                loop up1
                ret

htoa:                   ;o/p proc
            mov ecx,4
            mov edi,dispbuff
dup1:
            rol bx,4
```

```asm
        mov al,bl
        and al,0fh
        cmp al,09
        jbe skip
        add al,07h
skip: add al,30h
        mov [edi],al
        inc edi
        loop dup1

        dispmsg dispbuff,4
        ret
```