EXPERIMENT NO. 1

1. **AIM**: Write an 8086 assembly level program to perform:

(a) Multiplication of two bytes.

(b) Multiplication of two words.

2. APPARATUS REQUIRED: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4(a) Program of byte multiplication:

```
MOV     AL, BYTE 1; Load AL with byte 1

MOV     CL, BYTE 2

IMUL     CL              ; Multiply byte 1and byte 2

INT        3               ; Product in AX
```

4(b) Program of word multiplication:

```
MOV      AX, (MULTIPLICAND)   ; get one word

MOV      CX, (MULTIPLIER)        ; get the second word

MUL      CX                              ; multiply them

MOV      (PRODUCT), AX             ; store low word of result

MOV      (PRODUCT + 2), DX        ; store high word of result

 INT        3                               ; exit
```

EXPERIMENT NO. 2

1. AIM: Write a program in 8086 assembly language to obtain a packed BCD byte from two ASCII encoded digits.

2. APPARATUS REQUIRED: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. General Information:

This program produces a packed BCD byte from two ASCII encoded digits. The first ASCII digit (5) is located in AL register and the second ASCII (9) is located in the BL register. The result (packed BCD) is stored in the AL register.

5. Program:

```
MOV      AL, 35H      ; load first ASCII digit into AL

MOV      BL, 39H      ; load second ASCII digit into BL

AND      AL, OFH      ; mask upper four bits of first digit

AND      BL, OFH      ; mask upper four bits of second digit

MOV      CL, O4H      ; load CX for 4 rotates required

ROL      AL, CL       ; rotate AL 4 bit positions

ADD      AL, BL       ; combine nibbles, result in AL

INT         3         ; exit
```

EXPERIMENT NO. 3

1. **AIM**: Write an 8086 assembly level program to perform BCD operations.

2. **APPARATUS REQUIRED**: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4(a). BCD Multiplication program:

```
 MOV       AL, 5          ; AL = 00000101 = unpacked BCD 5

  MOV       BH, 9          ; BH = 00001001 = unpacked BCD 9

  MUL        BH             ; AL * BH, result in AX

  AAM                      ; AX = 00000000 00101101 = 002D H
                            ; AX = 00000100 00000101
                            ; Which is unpacked BCD for 45.
   INT          3
```

4(b). BCD Division program:

```
   MOV     AX, 60 H        ; AX = D607 unpacked BCD for 67 decimal

   MOV    CH, 09 H         ; CH = 09 H

   AAD                      ; adjust to binary before division
                            ; AX = 0043 = 43H = 67H decimal

  DIV       CH              ; divide AX by unpacked BCD in CH
                          ; AL = quotient = 07 unpacked BCD
                            ; AH = remainder = 04 unpacked BCD
                            ; PF = 0, SF =0 , ZF = 0
    INT           3
```

4(c). BCD subtraction program:

```
MOV            AL, 9 H        ; AL = 0011 1001 = ASCII 9
MOV             BL, 5H         ; BL = 0011 0101 = ASCII 5
SUB             AL, BL         ; (9-5) results:
                           ; AL = 0000 0100 = BCD 04  ; CF = 0

AAS                            ; results:
                           ; AL = 0000 0100 = BCD 04
                           ; CF = 0 no borrow required
INT                  3
```

EXPERIMENT NO. 4

1. **AIM**: Write an 8086 assembly level program that:
    (a) Scans a string of characters for “FF”.
    (b) Determines the end of string (EOS).

2. **APPARATUS REQUIRED**: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4(a) PROGRAM:

```
 MOV      AL, 0D H                 ; byte to be scanned for in AL

 MOV      DI, OFFSET TXT STR       ; offset of string to DI

 MOV      CX, 80 H                 ; CX used as element counter

 CLD                               ; clear DF so DI auto increments

REPNE      SCASB                   ; compare byte in string with byte
                                    ; In AL in a loop.
                                   ; If no match found CX will be 0,
                                    ; Else SI and DI will point to the
                                     Element after the first match,
INT        3
```

4(b) Modify 4(a) to determine end of string (EOS).

EXPERIMENT NO. 5

1. **AIM**: Write an 8086 assembly level program to perform 32 bit Division.

2. **APPARATUS REQUIRED**: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations.

4. **PROGRAM:**

```
     CMP      CX, 0H          ; check for illegal divide
      JE     ERROR EXIT   ; divisor = 0 so exit
      MOV    BX, AX          ; save lower order of dividend
     MOV     AX, DX          ; position high word for divide
     MOV     DX, 0000H    ; zero DX
     DIV       CX                 ; AX/CX, quotient in AX, remainder in DX
     MOV     BP, AX           ; save higher order of final result
     MOV     AX, BX           ; get back lower order of dividend
    DIV       CX                  ; AX / CX quotient in AX  ; remainder in DX
    MOV     CX, DX            ; pass remainder back in CX
    MOV     DX, BP            ; pass higher order result back in DX.
    CLC                               ; clear carry to indicate valid result
    JMP       EXIT               ; finished
ERROR-EXIT: STC                    ; set carry to indicate divide by zero
EXIT:     INT            3
```

EXPERIMENT NO. 6

1. AIM: Write an 8086 assembly level program to perform
case conversion of a string.

2. APPARATUS REQUIRED: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. PROGRAM:

```
            MOV     CX, 32                  ; no. of characters to change

            LEA     BX, TITLEX              ; first character to change

    B20
            MOV     AH, (BX)                ; character from TITLE

            CMP     AH, 61 H                ; is it

            JB      B30                     ; lower

            CMP     AH, 7A H                ; case

            JA      B30                     ; letter?

            AND     AH, 11011111B           ; yes- convert

            MOV     (BX), AH                ; restore in TITLEX

  B30
            INC     BX                      ; set for next character

            LOOP    B20                     ; loop for 32 times

            INT     3
```

EXPERIMENT NO. 7

1. AIM: Write an 8086 assembly level program to perform
BCD string addition.

2. APPARATUS REQUIRED: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. PROGRAM:

```
                CLC                                  ; no carry initially

                CLD                                  ; forward strings

                MOV    SI, OFFSET STRING-1           ; establish string pointers

                MOV    DI, OFFSET STRING-2

                MOV    CX, LEN-STR

                JCXZ    FINISH

CYCLE:          LODS    STRING-1                     ; get string-1 element

                ADC     AL, (DI)                     ; add string -2 element

                AAA                                  ; correct for ASCII

                STOS    STRING-2                     ; result into string -2

                LOOP   CYCLE                         ; repeat for entire element

FINISH:         INT        3
```

EXPERIMENT NO. 8

1. AIM: Write an 8086 assembly level program to perform ASCII to Binary conversion.

2. APPARATUS REQUIRED: MICROPROCESSOR 8086 KIT.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. PROGRAM:

```
            MOV        CX, 10             ; mult factors

            LEA        SI, ASCVAL-1      ; address for ASCVAL

            MOV BX,    ASCLEN             ; length of ASCVAL

B20:

            MOV        AL, (SI +BX)       ; select ASCII characters

           AND         AX, 000F           ; remove3-zone

           MUL         MULT 10            ; multiply by 10 factor

           ADD         BINVAL, AX         ; add to binary

           MOV         AX, MULT10         ; calculate next 10 factor

           MUL         CX

           MOV         MULT10, AX

           DEC         BX                 ; last ASCII character

          JNZ          B20                ; no continue

          INT          3
```

EXPERIMENT NO 9

1. AIM: Design an 8255 control word to configure 8255 in mode 0, i.e. simple input output mode. All the ports are in output mode. Write an assembly level program to transmit 55 H to Port A, AA H to Port B and CC H to Port C.

2. APPARATUS REQUIRED: 8086 microprocessor kit, 8255 interface module and 50 pins connecting cable.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. PROGRAM:

```
  MOV      AL, 80 H                 ; mode 0, all port in output mode

 OUT       CMD_PORT_55, AL

 MOV       AL, 55 H; data for port A

 OUT        PORTA_55, AL

 MOV       AL, 0AAH                 ; data for port B

OUT        PORTB_55, AL

MOV       AL, 0F                    ; data for port C

OUT       PORTC_55, AL

CALL SAVE_REG

JMP DISP_F_PRMT                     ; return control to monitor
```

EXPERIMENT NO 10

1. **AIM:** Write an 8086 assembly level program to configure 8253 counter 0 in mode 0, i.e. interrupt on terminal count. Write a program to Read / load lower 8 bits and then higher 8 bits of the counter.

2. **APPARATUS REQUIRED**: 8086 microprocessor kit, 8253 interface module and 50 pins connecting cable.

3. A suggestive program is provided for your reference. Please debug this program and performs correct operations

4. **PROGRAM;**

```
                 MOV      AL, 07 FH                  ; Unmask IRQ 7

                 OUT      OCW1, AL                   ; Send OCW1

                 STI                                 ; Enable interrupts

                MOV      AL, 30H                     ; Binary counter_0 selected,

                                                      ; Mode 0 read / loads LSB

                                                      ; First and then MSB.

                OUT      CMD_PORT_53,

                MOV      AL, 05H

                OUT      COUNTER_0, AL; COUNTER_0 LSB

                MOV      AL, 00H

                OUT      COUNTER_0, AL; COUNTER_0 MSB
B_1
                MOV      AL, 00H              ; Binary counter_0 mode 0, counter latch

               OUT       CMD_PORT_53, AL

               MOV       DL, AL

              IN         AL, COUNTER_0               ; Read MSB

              MOV        DH, AL

              JMP        B_1
```