Modified programs:

1.
Statement: Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

Sample problem

(4000H) = 19H
(400IH) = 6AH
(4004H) = I5H (4003H) = 5CH
Result = 6A19H - 5C15H = OE04H
(4004H) = 04H
(4005H) = OEH

Source program:

```
LHLD 4000H      : Get first 16-bit number in HL
XCHG            : Save first 16-bit number in DE
LHLD 4002H      : Get second 16-bit number in HL
MOV A, E            : Get lower byte of the first number
SUB L           : Subtract lower byte of the second number
MOV L, A            : Store the result in L register
MOV A, D            : Get higher byte of the first number
SBB H           : Subtract higher byte of second number with borrow
MOV H, A            : Store l6-bit result in memory locations 4004H and 4005H.
SHLD 4004H      : Store l6-bit result in memory locations 4004H and 4005H.
HLT             : Terminate program execution.
```

2.

Statement: Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H.
a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H.
b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H.
a. Sample problem

4200H = 04H
4201H = 10H
4202H = 45H
4203H = 33H
4204H = 22H
Result = 10 +41 + 30 + 12 = H
4300H = H

Source program:

```
LDA 4200H
MOV C, A          : Initialize counter
SUB A             : sum = 0
LXI H, 420lH      : Initialize pointer
BACK:  ADD M              : SUM = SUM + data
INX H             : increment pointer
DCR C             : Decrement counter
JNZ BACK          : if counter  0 repeat
STA 4300H         : Store sum
HLT               : Terminate program execution
```

3.

Statement:Write a program to count number of l's in the contents of D register and store the count in the B register.

```
 MVI B, 00H
        MVI C, 08H
        MOV A, D
   BACK: RAR
        JNC SKIP
        INR B
   SKIP: DCR C
        JNZ BACK
        HLT
```

4.

Statement:Write assembly language program to with proper comments for the following: To display decimal decrementing counter (99 to 00) at port 05 H with delay of half seconds between .each count. Write as well the delay routine giving delay of half seconds. Operating frequency of microprocessor is 3.072 MHz. Neglect delay for the main program.

Source Program:

```
    MVI C, 99H          : Initialize counter
BACK: MOV A, C
    ANI OF          : Mask higher nibble
    CPI OF
    JNZ SKIP
    MOV A, C
    SUI 06          : Subtract 6 to adjust decimal count
    MOV D, A
```

```
SKIP: MOV A, C
      OUT 05            : send count on output port
      CALL Delay              : Wait for 0.5 seconds
      DCR C                   : decrement count
      MOV A, C
      CPI FF
      JNZ BACK          : If not zero, repeat
      HLT                     : Stop execution

Delay subroutine:

Delay: LXI D, Count
Back: DCX D                   : 6 T-states
      MOV A, D          : 4 T-states
      ORA E                   : 4 T-states
      JNZ Back          : 10 T-states
      RET
```

Alias:

4.

Statement: Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

Sample problem

```
(4000H) = 19H
(400IH) = 6AH
(4004H) = I5H (4003H) = 5CH
Result    = 6A19H - 5C15H = OE04H
(4004H) = 04H
(4005H) = OEH
```

Source program:

```
LHLD 4000H          : Get first 16-bit number in HL
XCHG                : Save first 16-bit number in DE
LHLD 4002H          : Get second 16-bit number in HL
MOV A, E                  : Get lower byte of the first number
```

```
SUB L           : Subtract lower byte of the second number
MOV L, A              : Store the result in L register
MOV A, D              : Get higher byte of the first number
SBB H           : Subtract higher byte of second number with borrow
MOV H, A              : Store l6-bit result in memory locations 4004H and 4005H.
SHLD 4004H      : Store l6-bit result in memory locations 4004H and 4005H.
HLT             : Terminate program execution.
```

5. half wave rectifier 0/p

```
        MVI A, 80
        OUT 43
LOOP    LXI H, 8900
        MVI C, 07(NO OF POINTS)
LOOP1       MOV A, M
        OUT 40
        CALL DELAY1
        INX H
        DCR C
        JNZ LOOP1
        MVI A, 00
        OUT 40
        CALL DELAY
        JMP LOOP

DELAY  MVI D,FF
REPT   DCR D
        JNZ REPT
        RET

DELAY1      MVI D, 3A (IT'S ACTUALLY FF/NUMBER OF POINTS)
REPT1       DCR D
            JNZ REPT1
      RET;;;;;;
THE VALUES OF THE POINTS SHOULD BE ENTERED FROM 8900
;;;;;;THE VALUES SHOULD BE (SIN(F*n)/.02)
 IN HEX.    WHERE F = 180/NO OF POINTS(N),
```

6. full wave rectifier

```
            MVI A,80
            OUT 43
LOOP        LXI H,8900
            MVI C,07(NO OF POINTS)
LOOP1       MOV A,M
            OUT 40
            CALL DELAY1
```

```
        INX H
        DCR C
        JNZ LOOP1
        JMP LOOP


DELAY1  MVI D,3A (ITS ACTUALLY FF/NUMBER OF POINTS)
REPT1   DCR D
        JNZ REPT1
        RET

;;;;;;THE VALUES OF THE POINTS SHOULD BE ENTERED FROM 8900
;;;;;;THE VALUES SHOULD BE (SIN(F*n)/.02) IN HEX.   WHERE F = 360/NO OF
POINTS(N),
                                                    n = 0,1,2,...N-1
```

7.

Modified sine

```
        MOV A,80
        OUT 43
LOOP    LXI H,8900
        MVI C,07(NUMBER OF POINTS)
        MVI A,00
        OUT 40
        CALL DELAY
LOOP1       MOV A,M
        OUT 40
        CALL DELAY1
        INX H
        DCR C
        JNZ LOOP1
        MVI A,00
        OUT 40
        CALL DELAY
        JMP LOOP


DELAY   MVI D,FF
REPT    DCR D
        JNZ REPT
        RET


DELAY1  MVI D,3A (ITS ACTUALLY FF/NUMBER OF POINTS)
```

```
REPT1   DCR D
        JNZ REPT1
        RET
```

;;;;;;THE VALUES OF THE POINTS SHOULD BE ENTERED FROM 8900
;;;;;;THE VALUES SHOULD BE ((2.5 + SIN(F*n))/.02) IN HEX. WHERE F = 180/NO OF POINTS(N),

n = 0,1,2,...N-1

8.

Modified triangular

```
        MOV A,80
        OUT 43
LOOP    MVI A,00
        OUT 40
        CALL DELAY
LOOP1   XRA A
        OUT 40
        INR A
        CPI FF
        JNZ LOOP1

LOOP2   OUT 40
        DCR A
        CPI 00
        JNZ LOOP2

        MVI A,00
        OUT 40
        CALL DELAY
        JMP LOOP


DELAY   MVI D,FF
REPT    DCR D
        JNZ REPT
        RET
```

9.

12 e

```
        MOV A,80
        OUT 43
LOOP    MVI A,00
        OUT 40
        CALL DELAY
LOOP1   XRA A
        OUT 40
        INR A
        CPI 7F
        JNZ LOOP1

        MVI A,7F
        OUT 40
        CALL DELAY

        MVI A,FF
        OUT 40
        CALL DELAY


        MVI A,7F
        OUT 40
        CALL DELAY

LOOP2   OUT 40
        DCR A
        CPI 00
        JNZ LOOP2

        JMP LOOP



DELAY   MVI D,FF
REPT    DCR D
        JNZ REPT
        RET
```

10. qno.8 to divide it in to 4 section

```
        LXI H,8900
        MOV A,M
        ANI 0F
```

```
        MOV B,A
        MOV A,M
        ANI F0
        RRC
        RRC
        RRC
        RRC
        INX H
        MOV M,B
        INX H
        MOV M,A

;       WRITE THE MULTIPLICATION CODE FROM HERE, WITH DATA AT MEMORY
8901 AND 8902 AND STORE THE RESULT IN 8903
```