

Lab 8

AIM: Demonstrate the use of control flow, comparison and arithmetic instructions in 8085 Assembly Language to manipulate the data and implement algorithms for ordering values checking specific properties of numbers.

Programs

Question 1. A string of readings is stored in the memory location starting at 2070H and the end of the string is indicated by the byte 0DH. Write an 8085 assembly language program to check each byte in the string and save the bytes in the range of 30H to 39H (Both inclusive) in memory location starting from 2090H. Also count the number of bytes accepted from the string between 30H to 39H.

DATA(H): 35, 2F, 30, 39, 3A, 37, 7F, 31, 0D, 32

Answer:

- Assuming all the values are present in the proper memory locations.

```
; Initialization
LXI H, 2070H    ; Load memory address
LXI D, 2090H    ; Load destination address
MVI C, 00H      ; Initialize count to 0

LOOP:
    MOV A, M     ; Load byte from memory
    CPI 0DH      ; Check for end of string
    JZ DONE      ; If end, terminate
    CPI 30H      ; Compare with lower bound
    JC NEXT      ; Skip if below 30H
    CPI 3AH      ; Compare with upper bound +1
    JNC NEXT     ; Skip if greater than 39H
    MOV M, A     ; Store in destination
    INX D        ; Increment destination pointer
    INR C        ; Increment count

NEXT:
    INX H        ; Move to next byte
    JMP LOOP     ; Repeat loop

DONE:
    MOV M, C     ; Store count at the end
    HLT         ; Halt program
```

Question 2. Write an 8085 assembly language program to find the factorial of a given number.

Answer:

- Assuming all the values are present in the proper memory locations.

```

; Initialization
MVI B, 05H    ; Number to find factorial
MVI C, 01H    ; Initialize factorial result to 1

LOOP:
    MOV A, B    ; Copy B to A
    CPI 01H     ; Check if B is 1
    JZ DONE     ; If 1, terminate

    MOV D, C    ; Copy result to D
    MVI A, 00H  ; Clear A for multiplication
    MOV E, D    ; Copy multiplier to E

    MUL_LOOP:
        ADD D    ; Multiply A = A + D (loop-based multiplication)
        DCR E    ; Decrement multiplier
        JNZ MUL_LOOP ; Repeat until zero

    MOV C, A    ; Store new factorial result
    DCR B      ; Decrement B
    JMP LOOP    ; Repeat loop

DONE:
    HLT        ; Halt program

```

Question 3. Write an 8085 assembly language program to check whether a given number is prime or not. If the number is prime, store 01H at the memory location which stores the result, else 00H.

Answer:

- Assuming all the values are present in the proper memory locations.

```

; Initialization
MVI B, 07H    ; Example number (7)
MVI C, 02H    ; Start divisor from 2
MVI D, 01H    ; Assume number is prime (01H)

CHECK:
    MOV A, B    ; Copy number to A
    MOV E, C    ; Copy divisor to E

    ; Division logic begins
    MVI L, 00H  ; Clear remainder
    MOV H, 00H  ; Clear quotient
    MOV D, A    ; Copy dividend
    MOV A, E    ; Copy divisor

    DIV_LOOP:
        CMP D    ; Compare divisor and dividend
        JC DIV_DONE

```

```

    SUB E      ; Subtract divisor from dividend
    INX H      ; Increment quotient
    JMP DIV_LOOP

DIV_DONE:
    MOV L, D   ; Store remainder in L
    ; Division logic ends

    MOV A, L   ; Get remainder
    CPI 00H    ; Check if remainder is zero
    JZ NOTPRIME; If zero, not prime
    INR C      ; Increment divisor
    MOV A, C   ; Compare divisor with number/2
    MOV E, B
    RRC        ; Divide number by 2
    CMP C
    JNC CHECK  ; Continue loop if divisor < number/2

JMP PRIME

NOTPRIME:
    MVI D, 00H ; Set flag to not prime (00H)

PRIME:
    MOV M, D   ; Store result
    HLT        ; Halt program

```

Question 4. Write an assembly Language Program to find out the GCD (Greatest Common Divisor) for the numbers using the euclidean algorithm. DATA: (12,18)10 = 610 (12,17)10 = 110

Answer:

- Assuming all the values are present in the proper memory locations.

```

LXI H, 2070H ; Load address of first number
MOV B, M     ; Load first number
INX H        ; Move to next memory location
MOV C, M     ; Load second number

GCD_LOOP:
    CMP C     ; Compare B and C
    JZ DONE   ; If equal, GCD found
    JNC SWAP  ; Swap if B < C
    SUB C     ; B = B - C
    JMP GCD_LOOP

SWAP:
    MOV D, B
    MOV B, C
    MOV C, D
    JMP GCD_LOOP

```

DONE:

```
MOV M, B    ; Store GCD in memory
HLT         ; Halt program
```

Question 5. Write an 8085 Assembly Language Program to calculate the average of a set of 10 numbers

Answer:

- Assuming all the values are present in the proper memory locations.

```
LXI H, 2070H ; Load address of numbers
MVI C, 0AH   ; Set counter (10 numbers)
MVI B, 00H   ; Initialize sum to 0

SUM_LOOP:
    MOV A, M ; Load number
    ADD B    ; Add to sum
    MOV B, A ; Store result in B
    INX H   ; Move to next number
    DCR C   ; Decrement counter
    JNZ SUM_LOOP

    MOV A, B ; Load sum
    MVI C, 0AH ; Load divisor (10)
    CALL DIV ; Call division subroutine
    MOV M, H ; Store average in memory
    HLT     ; Halt program
```