Wap to add 2, 8-bit data and store result in 2050H (75H and 2AH)

- ⇨ LDA 75H;(MVI A,75H)
- ⇨ ADI 2AH
- ⇨ STA 2050H
- ⇨ HLT

WAP to add 16-bit data and store result at 2055H (No use of DAD)

- ⇨ LXI B, 16-bit data
- ⇨ LXI D, 16-bit data
- ⇨ MOV A, C
- ⇨ ADD E
- ⇨ STA 2055H;(MOV L, A)
- ⇨ MOV A, B
- ⇨ ADC D
- ⇨ STA 2056H (MOV H, A)
- ⇨ HLT

WAP to perform 8-bit subtraction

WAP to perform 16-bit subtraction

WAP to transfer 10 bytes of data stored in memory from 2050H to new address starting at 2070H

- ⇨ LXI B, 2050H
- ⇨ LXI D, 2070H
- ⇨ MVI H,0AH
- ⇨ Repeat: LDAX B
- ⇨ STAX D
- ⇨ INX B
- ⇨ INX D
- ⇨ DCR H
- ⇨ JNZ Repeat
- ⇨ HLT

WAP to transfer data from table 1 to table 2 by swapping lower and upper nibbles of each byte.

- ⇨ LXI H,2050H
- ⇨ LXI B,2070H
- ⇨ MVI D,0AH
- ⇨ Repeat: MOV A,M
- ⇨ RLC

- ⇨ RLC
- ⇨ RLC
- ⇨ RLC
- ⇨ STAX B
- ⇨ INX B
- ⇨ INX H
- ⇨ DCR D
- ⇨ JNZ Repeat
- ⇨ HLT

WAP to set MSB and reset MSB of each byte of Table 1 and store in Table 2

- ⇨ LXI H,2050H
- ⇨ LXI B,2070H
- ⇨ MVI D,0AH
- ⇨ Repeat: MOV A, M
- ⇨ ANI FEH
- ⇨ ORI 80H
- ⇨ STAX B
- ⇨ INX H
- ⇨ INX B
- ⇨ DCR D
- ⇨ JNZ Repeat
- ⇨ HLT

WAP to multiply 2, 8-bit data (product never exceeds 8-bit)

- ⇨ MVI B, 8-bit data
- ⇨ MVI D, 8-bit data
- ⇨ MVI A,00H
- ⇨ Repeat: ADD B
- ⇨ DCR D
- ⇨ JNZ repeat
- ⇨ HLT

WAP to multiply 2, 8-bit data (product may exceeds 8-bit)

- ⇨ MVI C, 8-bit
- ⇨ MVI B,00H

- ⇨ MVI D, 8-bit
- ⇨ LXI H, 0000H
- ⇨ Repeat: DAD B
- ⇨ DCR D
- ⇨ JNZ Repeat
- ⇨ HLT

WAP to divide 2, 8-bit data

- ⇨ MVI A, 8-bit data
- ⇨ MVI C, 8-bit data
- ⇨ MVI D,00H
- ⇨ Repeat: CMP C
- ⇨ JC Go
- ⇨ SUB C
- ⇨ INR D
- ⇨ JMP Repeat
- ⇨ Go: MVI E, A
- ⇨ HLT

WAP to transfer data from table 1 to table 2 by swapping D_5 and D_1 bits of each byte.

- ⇨ LXI H, 2050H
- ⇨ LXI B, 2070H
- ⇨ MVI D, 0AH
- ⇨ Repeat: MOV A, M
- ⇨ ANI 22H
- ⇨ CPI 22H
- ⇨ JZ Skip
- ⇨ CPI 00H
- ⇨ JZ Skip
- ⇨ MOV A, M
- ⇨ XRI 22H
- ⇨ JMP Go
- ⇨ Skip: MOV A, M
- ⇨ Go: STAX B
- ⇨ INX B
- ⇨ INX H
- ⇨ DCR D
- ⇨ JNZ Repeat
- ⇨ HLT

WAP to count no of 1’s in a byte of data

- MVI A, 8-bit data
- MVI B,00H
- MVI C,08H
- Repeat: RLC
- JNC Go
- INR C
- Go: DCR C
- JNZ Repeat
- HLT

WAP to count no of 0’s in a byte of data

WAP to count no of 0’s and 1’s in a byte of data

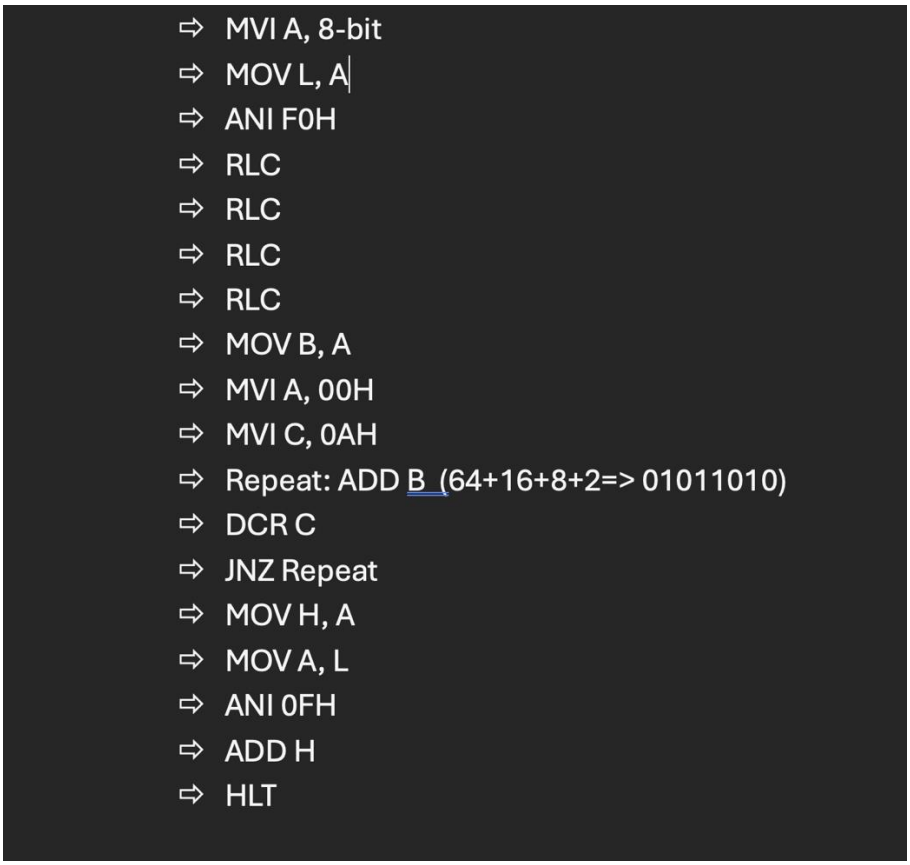
WAP to add corresponding bytes of two tables and store the result in third table

- LXI H, 2050H
- LXI D, 2070H
- LXI B, 2090H
- MVI A,14H
- Repeat: PUSH PSW; stack (a-> 14h and flags--)
- LDAX D
- ADD M
- STAX B
- INX H
- INX B
- INX D
- POP PSW; from stack(14h->a -----flags)
- DCR A
- JNZ Repeat
- HLT

WAP to convert BCD 8-bit into binary

- MVI A, 8-bit
- MOV E, A
- MVI B, 09H
- ANI F0H
- RLC
- RLC
- RLC
- RLC

- Repeat: ADD A; (64+16+8+2=> 01011010)
- DCR B
- JNZ Repeat
- MOV C, A
- MOV A, E
- ANI 0FH
- ADD E
- HLT



WAP to convert binary to BCD

- MVI A, 8-bit data
- MVI D, 00H
- FirstDiv: CPI C4H
- JC Terdiv
- SUB C4H
- INR D
- JMP FirstDiv
- Terdiv: MVI E, 00H

- ⇨ SecDiv: CPI 0AH
- ⇨ JC Terdiv2
- ⇨ SUB 0AH
- ⇨ INR E
- ⇨ JMP SecDiv
- ⇨ Terdiv2: MOV L, A
- ⇨ MOV A, E
- ⇨ RLC
- ⇨ RLC
- ⇨ RLC
- ⇨ RLC
- ⇨ ADD L
- ⇨ MOV E, A
- ⇨ HLT

WAP to count no of data less than or equal to 45H in a table.

- ⇨ LXI H, 2050H
- ⇨ MVI C, 0AH
- ⇨ MVI B, 00H
- ⇨ MVI D, 00H
- ⇨ Repeat: MOV A, M
- ⇨ CPI 45H
- ⇨ JNC Incless
- ⇨ INR B
- ⇨ Incless: CPI 45H
- ⇨ JNZ Incequal
- ⇨ INR D
- ⇨ Incequal: INX H
- ⇨ DCR C
- ⇨ JNZ Repeat
- ⇨ MVI A, B
- ⇨ ADD D
- ⇨ HLT

WAP to add lower and upper nibble of a data and put in another table

- ⇨ LXI H,2050H
- ⇨ LXI D, 2070H
- ⇨ MVI C, 0AH
- ⇨ Repeat: MOV A, M
- ⇨ ANI F0H

- ⇨ RLC
- ⇨ RLC
- ⇨ RLC
- ⇨ RLC
- ⇨ MOV B, A
- ⇨ MOV A, M
- ⇨ ANI 0FH
- ⇨ ADD B
- ⇨ STAX D
- ⇨ DCR C
- ⇨ JNZ Repeat
- ⇨ HLT