1. Write a program to convert 8-bit Gray code into its equivalent binary number.

Code:

```
LXI H, 2100H
MOV A, M
MVI B, 07H
back: RAR
XRA M
DCR B
JNZ back
INX H
MOV M, A
HLT
```

2. Write a program to scan an 8-bit number from a block of 10 bytes and store the count after the block

Code:

```
LXI H, 2100H
MVI E, 00H
MOV B, M
INX H
MOV C, M
back: INX H
MOV A, M
CMP B
JZ loop
DCR C
JNZ back
JMP exit
loop: INR E
DCR C
JNZ back
exit: INX H
MOV M, E
HLT
```

2100: Target Element
2101: Length of the array, in this problem it is 10 □ 0Ah
2102: First Element of the Array

Example: 2100: 45h, Target Element
2101: 0A, Length of the array
2102 to 210B: 34,56,23,12,56,45,76,45,34,12

Result: 210C: 2

3. Write a program to multiply 16-bit number

Code:

```
LXI B,0000H
LHLD 2100H
SPHL
LHLD 2102H
XCHG
back: DAD SP
JNC skip
INX B
skip: DCX D
MOV A, E
ORA D
JNZ back
SHLD 2105H
MOV A, C
STA 2107H
MOV A, B
STA 2108H
HLT
```

4. Write a program to divide a 16-bit number

Code:

```
LXI B,0000H
LHLD 2102H; Divisor
XCHG
LHLD 2100H; Dividend
back: MOV A, L
SUB E
MOV L, A
MOV A, H
SBB D
MOV A, H
JC skip
INX B
JMP back
skip: DAD D
SHLD 2106H
MOV L, C
MOV H, B
```

```
SHLD 2104H
HLT
```

5. Write a program to calculate sum of series of number

Code:

```
LXI H,2100H
MOV B, M; First Term ‘a’
LXI H,2101H
MOV D, M; Common Difference ‘d’
LXI H,2102H
MOV E, M; Number of Terms ‘n’
MVI C,00H
MOV H, B
DCR E
back: MOV A, B
ADD D
MOV B, A
MOV A, H
ADD B
JNC skip
INR C
skip: MOV H, A
DCR E
JNZ back
STA 2103H
MOV A, C
STA 2104H
HLT
```

6. Write a program to find out square of given array of number

Code:

```
LXI H,2100H
MVI D,0AH
back1: XRA A
MOV B, M
MOV C, M
back2: ADD B
DCR C
JNZ back2
MOV M, A
INX H
DCR D
JNZ back1
HLT
```

7. Write a program to find out cube of given array of number

Code:

```
LXI H,2100H
MVI E,0AH
L3: XRA A
MOV A, M
MOV B, M
MOV C, M
MOV D, M
DCR B
L1: ADD D
DCR B
JNZ L1
DCR C
JNZ L2
MOV M, A
INX H
DCR E
JNZ L3
HLT
L2: MOV B, M
JMP L1
```

8. Write a program to convert BCD number into equal hexadecimal number.

Code:

```
LDA 2100H
MOV B, A
ANI 0FH
MOV C, A
MOV A, B
ANI F0H
RRC
RRC
RRC
RRC
MOV D, A
MVI E,0AH
XRA A
back: ADD D
DCR E
JNZ back
ADD C
```

```
STA 2101H
HLT
```

9. Write a program to find out square root of a given data.

Code:

```
LDA 2100H
MVI D,01H
MVI E,01H
back1: SUB D
JZ back2
INR D
INR D
INR E
JMP back1
back2: MOV A, E
STA 2101H
HLT
```

10. Write a program to display an element of a Fibonacci series.

Code:

```
LXI H, 2100H
LDA 2100H
MOV D, A
MVI B, 00H
MVI C, 01H
MVI A, 00H
back: MOV M, A
ADD C
MOV B, C
MOV C, A
MOV A, B
INX H
DCR D
JNZ back
HLT
```

11. Write a program to generate square waveform of frequency 10 KHz using SOD.

Code:

```
MVI A, 40H
SIM
back: MVI A, C0H
SIM
```

```
CALL delay0.05ms
MVI A,40H
SIM
CALL delay0.05ms
JMP back

delay0.05ms: MVI B, 03H
             back1: DCR B
             JNZ back1
             RET
```

12. Write a program to sort out positive and negative numbers.

Code:

```
LXI H, 2100H
LXI D, 2200H
MVI C, 0AH
back: MOV A, M
RAL
JC skip
MOV A, M
STAX D
INX D
skip: INX H
DCR C
JNZ back

LXI H, 2100H
LXI D, 220AH
MVI C, 0AH
back2: MOV A, M
RAL
JNC skip2
MOV A, M
STAX D
INX D
skip2: INX H
DCR C
JNZ back2

HLT
```

13. Write a program to find out factorial of given array of number.

Code:

```
LXI H, 2100H
MOV B, M
MVI D, 01H
back1: CALL multi
DCR B
JNZ back1
INX H
MOV M, D
HLT

multi: MOV E, B
       XRA A
       loop: ADD D
       DCR E
       JNZ loop
       MOV D, A
       RET
```

14. Write a program to calculate series of even number.

Code:

```
MVI C, 0AH
MVI B, 00H
LXI H, 2100H
back: MOV A, M
RRC
JC odd
MOV A, B
ADD M
MOV B, A
odd: INX H
DCR C
JNZ back
MOV M, B
HLT
```

15. Write a program to calculate the series of odd number.

Code:

```
MVI C, 0AH
MVI B, 00H
```

```
LXI H, 2100H
back: MOV A, M
RRC
JNC even
MOV A, B
ADD M
MOV B, A
even: INX H
DCR C
JNZ back
MOV M, B
HLT
```

16. Write a program to perform 2`s compliment of a number. Write for 10 nos

Code:

```
LXI H, 2100H
MVI C, 0AH
back: MOV A, M
CMA
ADI 01H
MOV M, A
INX H
DCR C
JNZ back
HLT
```

17. Write a program to output contents of B register LSB to MSB on the SOD pin.

Code:

```
MVI C, 08H
MOV A, B
back: RRC
MOV B, A
JNC skip
MVI A, COH
SIM
JMP next
skip: MVI A, 40H
SIM
next: CALL delay
DCR C
JNZ back
HLT

delay: LXI D, FFFFH
back1: DCX D
```

```
MOV A, D
ORA E
JNZ back1
RET
```

18. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.Add two 4-digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.

Code:

Packing of BCD Numbers

```
LDA 4201H
RLC
RLC
RLC
RLC
ANI F0H
MOV C, A
LDA 4200H
ADD C
STA 4300H
HLT
```

Addition of two 4-digit BCD

```
LXI H, 3000H
LXI D, 5000H
DAD D
MOV A, L
DAA
STA 2300H
MOV A, H
DAA
STA 2301H
HLT
```

19. Subtract the BCD number stored in E register from the number stored in the D register

Code:

```
MVI E, 27H
MVI A, 99H
SUB E
INR A
MVI D, 81H
ADD D
DAA
STA 2100H
HLT
```

20. Write an assembly language program to multiply 2 BCD numbers

Code:

```
LXI H, 2100H
MOV B, M
INX H
MOV C, M
MVI E, 00H
MOV H, E
MOV A, E
CMP C
JZ done
loop: ADD B
DAA
MOV D, A
JZ skip
MOV A, H
ADI 01H
DAA
MOV H, A
skip: MOV A, E
ADI 01H
DAA
MOV E, A
CMP C
MOV A, D
JNZ loop
done: MOV L, A
SHLD 2200H
HLT
```

21. Program to search a character in a given string.

Code:

```
LXI H, 2100H
MVI E, 00H
MOV B, M
INX H
MOV C, M
back: INX H
MOV A, M
CMP B
JZ loop
DCR C
JNZ back
JMP exit
loop: INR E
DCR C
JNZ back
exit: INX H
MOV M, E
HLT
```

22. A Block of 10 Bytes is present in memory from 2100H onwards. WAP to mask 3 MSB's and 2 LBS's and store the result from 2200H onwards.

Code:

```
LXI H, 2100H
LXI B, 2200H
MVI E, 0AH
back: MOV A, M
ANI 1CH
STAX B
INX H
INX B
DCR E
JNZ back
HLT
```

23. A Block of 10 Bytes is present in memory from 2100H onwards. WAP to reset 4 LSB's and store the result from 2200H onwards.

Code:

```
LXI H, 2100H
LXI B, 2200H
MVI E, 0AH
```

```
back: MOV A, M
ANI F0H
STAX B
INX H
INX B
DCR E
JNZ back
HLT
```

24. WAP to perform the addition of 10 no. present in memory from address 2100H onwards. Store the result of addition into memory.

Code:

```
LXI H, 2100H
MVI C, 0AH
XRA A
STC
CMC
back: MOV B, M
ADC B
INX H
DCR C
JNZ back
MOV M, A
INX H
MVI A, 00H
ADC A
MOV M, A
HLT
```

25. WAP to perform addition of two 64-bit no. and store the result of addition into memory from 2300H onwards. The two no. are present in memory from 2100H onwards.

Code: LXI H,21OOH

LXI B,2108H

LXI D,2300H

STC

CMC

UP:LDAX B

ADC M

STAX D

INX H

INX B

INXD

MOV A,L

CPI 08H

JNZ UP

RST 1

26. WAP to scan for 40H if 40H is present. Store the address of memory containing 40H into stack memory at 02FAH.

    Code:

    LXI H, 2100H
    LXI SP, 02FA
    MVI C, 0AH
    MOV B, M
    back: INX H
    MOV A, M
    CMP B
    JZ skip
    DCR C
    JNZ back
    skip: PUSH H
    HLT

27. Two Block of 10 Bytes is present in memory from 2100H and 2200H onwards. WAP to find how many mismatch bytes are present into these blocks and store the count at 2250H.

Code:

LXI H, 2100H
LXI D, 2200H
MVI C, 0AH
MVI B, 00H
back: LDAX D
CMP M
JNZ skip
INR B

skip: INX H
INX D
DCR C
JNZ back
MOV A, B
STA 2250H
HLT

28. A Block of 10 Bytes is present in memory from 2100H onwards. WAP to find even and odd no. and store them from 2250H and 2200H onwards.

Code:

```
LXI H, 2100H
LXI D, 2200H
MVI C, 0AH
back: MOV A, M
ANI 01H
JZ skip
MOV A, M
STAX D
INX D
skip: INX H
DCR C
JNZ back

LXI H, 2100H
LXI D, 2250H
MVI C, 0AH
back2:MOV A, M
ANI 01H
JNZ skip2
MOV A, M
STAX D
INX D
skip2: INX H
DCR C
JNZ back2

HLT
```

29. WAP to scan for 80H. Store the count of no. of 80H present in the series at 2200H.

Code:

```
LXI H, 2100H
MVI E, 00H
```

```
MOV B, M
INX H
MOV C, M
back: INX H
MOV A, M
CMP B
JZ loop
DCR C
JNZ back
JMP exit
loop: INR E
DCR C
JNZ back
exit: MOV A, E
STA 2200H
HLT
```

30. Write a to scan for AA H. Store the address of memory location containing AA H as last found into the stack memory from 2738H.

Code:

```
LXI H, 2100H
LXI SP, 2738H
MVI C, 0AH
MOV B, M
back: INX H
MOV A, M
CMP B
JZ skip
DCR C
JNZ back
skip: PUSH H
HLT
```

31. WAP to solve the equation Y = mx+c where m, x and c are 8-bit no. present in memory from 2100H.

Code:

```
LXI H, 2100H
MOV B, M; value of 'm' of the equation
INX H
MOV C, M; value of 'x' of the equation
INX H
MOV D, M; value of 'c' of the equation
CALL multi
ADD D
```

```
INX H
MOV M, A
HLT

multi: MVI A, 00H
loop: ADD B
DCR C
JNZ loop
RET
```

32. A series is present in memory from address 2101H onwards. The length of the series is present at 2100H. WAP to find smallest no. and store at 2400H.

Code:

```
LXI H, 2100H
MVI C, M
LXI H, 2101H
MVI A, FFH
back: CMP M
JC skip
MOV A, M
skip: INX H
DCR C
JNZ back
STA 2400H
HLT
```

33. A Block of 10 Bytes is present in memory from address 2100H onwards. WAP to convert the no. into Excess-3 code and store the result into memory from 2200H onwards.

Code:

```
LXI H, 2100H
LXI D, 2200H
MVI C, 0AH
back: MOV A, M
ADI 03H
STAX D
INX D
INX H
DCR C
JNZ back
HLT
```

34. 10 Bytes are present in memory from 2100H onwards. WAP to find out no. of 1's present in these 10 Bytes and store the count at the end of the series.

Code:

```
LXI H, 2100H
MVI D, 00H
MVI B, 0AH
back2: MOV A, M
MVI C, 08H
back1: RAR
JNC skip
INR D
skip: DCR C
JNZ back1
INX H
DCR B
JNZ back2
MOV M, D
HLT
```

35. A Block of 10 Bytes is present in memory from address 2200H onwards. WAP to count no. of 1's in each Byte and store the count from 2100H onwards.

Code:

```
LXI H, 2200H
MVI C, 0AH
back2: MOV A, M
MVI B, 00H
MVI D, 08H
back1: RRC
JNC skip
INR B
skip: DCR D
JNZ back1
MOV M, B
INX H
DCR C
JNZ back2

LXI B, 2200H
LXI D, 2100H
MVI L, 0AH
back3: LDAX B
STAX D
INX B
INX D
DCR L
```

```
JNZ back3
HLT
```

36. Two Block of 10 Bytes are present in memory from 2100H and 2200H onwards. WAP to reverse the Byte and store from 2200H onwards.

Code:

```
LXI H, 2100H
LXI B, 2200H
MVI E, 0AH
back: MOV A, M
RRC
RRC
RRC
RRC
STAX B
INX B
INX H
DCR E
JNZ back
HLT
```

37. WAP to solve the equation Y = AB + C where A, B, C are 8 bits no. present into memory.

Code:

```
LXI H, 2100H
MOV B, M; value of 'A' of the equation
INX H
MOV C, M; value of 'B' of the equation
INX H
MOV D, M; value of 'C' of the equation
CALL multi
ADD D
INX H
MOV M, A
HLT

multi: MVI A, 00H
loop: ADD B
DCR C
JNZ loop
RET
```

38. WAP to solve the equation Y = A*B/C where A, B, C are 8 bits no. present into memory.

Code:

```
LXI H, 2100H
MOV B, M; value of 'A' of the equation
INX H
MOV C, M; value of 'B' of the equation
INX H
MOV D, M; value of 'C' of the equation
CALL division
CALL multi
INX H
MOV M, A
HLT

division: MVI E, 00H; Quotient
MOV A, C
loop: SUB D
INR E
CMP D
JNC loop
RET

multi: MVI A, 00H
loop2: ADD E
DCR B
JNZ loop2
RET
```

39. A series of no. is present in memory from 2201H onwards and length of series is present at 2200H. WAP to store the series in reverse order from 2250H.

Code:

```
LXI B, 2201H
LXI D, 2250H
LXI H, 2200H
MOV L, M
back: LDAX B
      STAX D
      INX B
      DCX D
      DCR L
      JNZ back
      HLT
```

40. WAP to find the 2's complement of 10 no. stored at memory location 2200H and store the complemented no. at 2100H onwards.

Code:

```
LXI H, 2200H
LXI D, 2100H
MVI C, 0AH
back: MOV A, M
CMA
ADI 01H
STAX D
INX D
INX H
DCR C
JNZ back
HLT
```